IJCLA **ISSN 0976-0962**

International Journal of Computational Linguistics and Applications

Vol. 5 **No. 2** **Jul-Dec 2014**

Editor-in-Chief
Alexander Gelbukh

ISSN 0976-0962

International Journal of Computational Linguistics and Applications

Vol. 5 **No. 2** **Jul-Dec 2014**

International Journal of Computational Linguistics and Applications – IJCLA (started in 2010) is a peer-reviewed international journal published twice a year, in June and December. It publishes original research papers related to computational linguistics, natural language processing, human language technologies and their applications.

The views expressed herein are those of the authors. The journal reserves the right to edit the material.

Indexing: Cabell's Directory of Publishing Opportunities.

Editor-in-Chief: Alexander Gelbukh

Subscription: India: Rs. 2699
Rest of the world: US$ 249

Payments can be made by Cheques/Bank Drafts/International Money Orders drawn in the name of BAHRI PUBLICATIONS, NEW DELHI and sent to:

BAHRI PUBLICATIONS
1749A/5, 1st Floor, Gobindpuri Extension,
P. O. Box 4453, Kalkaji, New Delhi 110019
Telephones: 011-65810766, (0) 9811204673, (0) 9212794543
E-mail: bahrius@vsnl.com; bahripublications@yahoo.com
Website: http://www.bahripublications.com

Printed & Published by Deepinder Singh Bahri, for and on behalf of **BAHRI PUBLICATIONS**, New Delhi.

International Journal of Computational Linguistics and Applications

Vol. 5 **No. 2** **Jul-Dec 2014**

CONTENTS

Editorial

This issue of IJCLA presents papers on lexical resources, text features, named entity recognition, detection of lexical functions, information extraction, learning concept identification, opinion mining, and speech processing.

J. M. Torres-Moreno *et al*. (France, Canada, Mexico, and Germany) present a corpus of German texts for similarity detection. Similarity detection between texts is a basic task underlying many natural language processing tasks such as information retrieval, text classification, opinion mining, and text summarization, to name only a few. The authors introduce a valuable lexical resource designed to train and evaluate algorithms for measuring text similarity.

G. Sidorov (Mexico) discusses various types of syntactic n-grams. Syntactic n-grams can be used as features to characterize texts instead of individual words or usual n-grams. Individual words are insufficient for describing meaning or sentiment carried by the text: say, *large screen* may carry positive sentiment when speaking about a phone, while individual words *large* and *screen* do not carry any sentiment. The technique of syntactic n-grams permits to capture long-distance relationships between words, such as *large screen* in *the phone has a large and brightly backlit screen*.

S. Zhou et al. (Japan) present a study of linguistic features useful for the task of named entity disambiguation. Different named entities can share the same name: for example, there is a number of different cities named *New York* or *Moscow*, *St. Andrew* may refer to a person or a university, etc. This justifies the need for algorithms capable of disambiguating such expressions in specific contexts. The authors discuss linguistic features of the texts that help in measuring local coherence of the text; choosing the candidate variant that maximizes the local coherence is a good heuristic for named entity disambiguation.

O. Kolesnikova (Mexico) considers the task of detecting lexical functions in verb-noun collocations in Spanish texts. Verb-noun collocations are pairs of words such as *make a decision*, *meet resistance*, or *keep a secret*. Lexical function in such a word pair, called collocation, is a very general meaning that the verb expresses with a specific noun.

In our examples, the meaning of *make* in *make* decision is, roughly, to indicate that the agent does what the noun expresses; the meaning of *meet* in *meet resistance* is to indicate that the subject undergoes what the noun expresses; the meaning of *keep* in *keep secret* is that the agent does what is expected to do with what the noun expresses. For many text processing tasks it is important to distinguish between such general meanings. The author discusses a technique for their automatic prediction.

M. F. Abdel Hady *et al*. (Egypt) suggest an unsupervised training method for information extraction using aligned bilingual corpora. Usual techniques for information extraction, and in particular for named entity recognition, require expensive lexical resources manually annotated by trained experts. Active learning reduces the amount of training examples by requesting the annotator to annotate a specific example, the most informative one at each step. In this paper the authors show how bilingual corpora can eliminate at all the need in specific manual annotation for the information extraction task, while active learnings significantly speeds up the process.

R. Piryani *et al*. (India) consider the task of identification of the learning concepts in educational texts. The system they present helps students using three-step approach. First, the system identifies the learning needs of a specific student. Next, the system retrieves the relevant learning materials and ranks them according to the suitability for the needs of the learner. Finally, the system presents the learning material to the student and monitors the learning process. The system combines methods from a number of computer science disciplines, such as natural language processing and information retrieval, recommender systems, and educational psychology, among others.

I. Cruz *et al*. (Mexico) describe a publicly available corpus for aspect-based opinion mining that they developed and a supervised machine learning method for identifying opinion aspects in texts that uses this corpus for training. Opinion mining is a fast-growing and very popular area of text processing. An important part of opinion mining process is identifying polarity of a text: whether, say, a user review of a phone Galaxy X expresses positive or negative opinion. However, it is common that people express opinions not about the product in general but of some its aspects: say, the user likes the screen of the phone but does not like its battery. Aspect-based opinion mining operates at this granularity level. While explicit aspects correspond to words present in the document, such as in *the screen is large* (the aspect is *screen*), implicit aspects are implied by other words but not mentioned explicitly, such as in the phone is inexpensive (the aspect is *price*). The authors present a

manually annotated corpus, the first of its kind, that gives such implicit aspects along with the words that imply them (indicators).

The last two papers of the volume are devoted to speech processing: one to speech synthesis and the other to speech recognition.

L.-Q. Tran *et al.* (Vietnam) address prosodic characteristics of airport announcements in Vietnamese. Generating correct prosodic information is important for the generated speech to sound naturally and to carry the correct sentiment. Vietnamese language is a tonal language, in which the role of stress is very different from the role of stress in European languages such as English or French. The authors present a description of stress and other prosodic features in airport announcements in Vietnamese and describe a Hidden Markov Model-based text-to-speech system in this domain that aims to correctly model this information.

C. Lhioui *et al*. (Tunisia) give a detailed analysis of the task of recognition of spontaneous speech in Arabic. The structure of spontaneous speech is quite different from that of written text. While semantic analysis of written text is a common research subject in natural language processing, much fewer effort has been dedicated to understanding spontaneous speech, which can be attributed to the fact that this task is much more difficult. The authors describe in detail their method, which combines two very different approaches: syntax-based approach and stochastic semantic decoding, which improves the robustness of the system. The authors discuss in detail specific difficulties of Arabic language and Arabic spontaneous speech. They also present a corpus that they developed for the task.

This issue of IJCLA will be useful for researchers, students, software engineers, and general public interested in natural language processing and its applications.

ALEXANDER GELBUKH
EDITOR IN CHIEF

MEMBER OF MEXICAN ACADEMY OF SCIENCES,
RESEARCH PROFESSOR AND
HEAD OF THE NATURAL LANGUAGE PROCESSING LABORATORY,
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN,
INSTITUTO POLITÉCNICO NACIONAL, MEXICO
WEB: <WWW.GELBUKH.COM>

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 9–24
Received 7/08/2014, accepted 12/09/2014, final 9/12/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

A German Corpus for Similarity Detection Tasks

JUAN-MANUEL TORRES-MORENO,[1,2]
GERARDO SIERRA,[1,3] AND PETER PEINL[4]

[1] *Université d'Avignon et des Pays de Vaucluse, France*
[2] *École Polytechnique de Montréal, Canada*
[3] *Universidad Nacional Autónoma de México, Mexico*
[4] *University of Applied Sciences Fulda, Germany*

ABSTRACT

Text similarity detection aims at measuring the degree of similarity between a pair of texts. Corpora available for similarity detection are designed to evaluate the algorithms to assess the paraphrase level among documents. In this paper we present a textual German corpus for similarity detection. The purpose of our corpus is to automatically assess the similarity between a pair of texts and to evaluate different similarity measures, both for whole documents or for individual sentences. Therefore we have calculated several simple measures on our corpus based on a library of similarity functions.

1 INTRODUCTION

Text similarity is a condition or property that can be measured between two or more texts, which determines the degree of similarity between them. Text similarity ranges between 0% (no relationship at all) and 100% (documents are identical). Also note that two similar texts do not need to share the content, neither verbatim nor expressed in other words. They may just cover the same topic or merely be written in the same language.

Similarity detection has been intensively studied and is of great interest for different applications of Natural Language Processing (NLP),

such as plagiarism and paraphrasing detection, fraud analysis, document clustering, machine translation, automatic text summarization and information retrieval.

To develop systems for similarity detection both a training and a test corpus, built to the requirements of the task to achieve, have to be available. For paraphrase detection, the corpus in particular must comprise the text source and the text paraphrasing the content of the text source. Corpora specifically designed for this task already exist, such as the METER Corpus,[5] the Microsoft Research Paraphrase Corpus,[6] and the PAN Plagiarism Corpus.[7] However, as they do not fulfil the requirements of a tool we are still working on, we needed an *ad hoc* corpus.

We are aiming at the assessment of the similarity between a pair of documents, not necessarily paraphrase detection. For that purpose we needed a gold-standard comparable corpus containing source texts and texts similar to them, either because they are paraphrases or because they just deal with the same topic. Even more, because they share the lexical units although do not share the topic[8]. Also we aim at a more precise assessment of similarity and at a mapping between the source text and the paraphrased text at the paragraph level.

The purpose of the paper is to present the methodology of the construction of a paraphrasing corpus, the description of the German corpus using this methodology, and an illustration of its usefulness with respect to standard simple measures for paraphrasing detection.

The paper is organized as follows. In Section 2 we give an overview of similarity detection and the current corpora for similarity detection. Then, in Section 3 we outline the usual simple measures to detect and evaluate similarity. Next, in Section 4 we describe the methodology to build our corpus. In Section 5 and Section 6 we report on the application and exploitation of different simple measures on our corpus, before concluding in Section 7.

2 SIMILARITY DETECTION

Similarity as a concept has a wide range of applications in different areas. Similarity implies different features and relationships among objects.

[5] http://nlp.shef.ac.uk/meter/

[6] http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/

[7] http://www.webis.de/research/corpora

[8] That issue will not be presented in this paper.

Depending on the area, context or perspective, similarity between objects can differ. Similarity in fact depends on the context surrounding the object. An object a is similar to b only referring to a context c [1]. Every task involving similarity must therefore specify the context and the features to focus on. For example, two books on the same library shelf are likely to be similar due to the same thematic, even if the content or the language is different.

Hence, similarity detection aims at comparing different objects and to observe the common features they share according to certain parameters. The units of language to compare in the context of NLP might be words, sentences, paragraphs or documents.

Text similarity ranges between the paraphrase of a sentence or paragraph from another document and a complete copy of a document. As [2] explain, there is a similarity spectrum from plagiarism (nearly identical documents or even identical documents) to topical similarity, passing through text reuse. In addition, two documents may be similar without any direct relationship, but by their similarity to a third text. For example, different newswires derived from a common source text provided by a news agency are similar, as are the homeworks of pupils on a common thematic or reviews or adaptations of a literary work.

2.1 *Existing Corpora for Similarity Detection*

Most of the well-known corpora on similar texts are designed to evaluate the algorithms to detect paraphrases among documents. Some comparable corpora of considerable size have been created automatically by using certain heuristics.

The METER corpus is a corpus of news texts collected manually from a news agency and nine daily newspapers [3, 4] that reuse these newswires. The resulting 1717 texts were manually classified at the document level into three categories, according to the relatedness of the texts to the original newswire: wholly derived (WD), if the note derives fully from the agency; partially derived (PD), if the article uses other sources besides the information provided by the agency; and non-derived (ND), if the note is written independently from the newswire provided by the agency. At the phrasal level, individual words and phrases were compared to find verbatim text, paraphrased text or none at all.

The Microsoft Research Paraphrase Corpus consists of 5801 pairwise aligned sentences that exhibit lexical and/or structural paraphrase alternations extracted from news reports [5]. It was created automatically using

string edit distance and discourse-based heuristic extraction techniques. The corpus has binary statements indicating whether human evaluators considered the pair of sentences to be semantically equivalent or not [6].

The PAN Plagiarism Corpus is a corpus for the evaluation of automatic plagiarism detection algorithms. For the source documents, texts of artificial plagiarism were created automatically through a heuristic of changing some parameter, such as document length, suspicious-to-source ratio, plagiarism percentage and plagiarism length, plagiarism languages and plagiarism obfuscation [7].

3 MEASURING TEXT SIMILARITY

Metrics for similarity detection assess either the commonality or the difference between two sets of data. The higher the commonality between two objects, the more similar they are. On the other hand, the higher the difference between two objects, the lower is their similarity. Hence, similarity increases with commonality, but decreases with difference [8].

The metrics calculate a score that can be normalized to be between zero and one. The ranking score is useful for different tasks, such as information retrieval or Question-Answering (Q&A) systems. However, for paraphrase detection purposes, a binary result is considered [9], but the similarity measures get a grade as result. Based on a threshold it is determined whether the compared texts are the same, a paraphrase or different [10].

There are three main approaches to similarity detection. They are based either on vector space models (term-based), on text alignment (linguistic knowledge-based) or on n-gram overlapping (string-based).

3.1 *Vector Space Models*

Vector Space Models are one of the simplest and most common way to assess content similarity among documents, which are considered as a bag of words. Therefore, words are supposed to appear independently while the order is irrelevant. A text is transformed into a term vector representation, following the removal of stop words and stemming. We focus on three metrics to determine the commonality between two texts (Cosine similarity, Dice similarity and Jaccard similarity), as well as on two metrics to measure dissimilarity (Euclidean distance and Manhattan distance) [11].

Cosine similarity is one of the most popular vector based similarity measures. A text is transformed into a vector space, so that the Euclidean cosine rule can be used to determine similarity. Cosine similarity between documents D_1 and D_2 in a vector space is defined as:

$$sim_C(\overrightarrow{D_1}, \overrightarrow{D_2}) = (\overrightarrow{D_1}, \overrightarrow{D_2}) = \sum_{j=1}^{k} w_{1,j} w_{2,j}.$$

The $w_{x,y}$ are the weight of the words calculated as the term frequency tf and k corresponds to the number of different terms.

Dice similarity uses the Dice coefficient, i.e. the ratio of twice the number of shared terms in the compared texts to the total number of terms in both texts. m_c is the number of common words in documents D_1 and D_2, and m_1 and m_2 the number of words of D_1 and D_2, respectively, Dice similarity is:

$$sim_D(\overrightarrow{D_1}, \overrightarrow{D_2}) = \frac{2m_c}{m_1 + m_2}.$$

Jaccard similarity measures similarity by comparing the number of common terms to the number of all unique terms in both texts. m_c being the number of common words between documents D_1 and D_2, and m_1 and m_2 the number of words of D_1 and D_2, respectively, Jaccard similarity is:

$$sim_J(\overrightarrow{D_1}, \overrightarrow{D_2}) = \frac{m_c}{m_1 + m_2 - m_c}.$$

Euclidean distance is an ordinary measure in the vector space model to determine the distance between the vector inputs, rather than the angle as in the cosine rule. Euclidean distance is defined as:

$$dist_E(\overrightarrow{D_1}, \overrightarrow{D_2}) = \|\overrightarrow{D_1}, \overrightarrow{D_2}\| = \sqrt{\sum_{j=1}^{k} (w_{1,j} - w_{2,j})^2}.$$

Manhattan distance can be described in two dimensions with discrete-valued vectors, where the distance value is simply the sum of the differences of their corresponding vectors:

$$dist_M(\overrightarrow{D_1}, \overrightarrow{D_2}) = \sum_{j=1}^{k} (w_{1,j} - w_{2,j}).$$

3.2 *Text Alignment*

Unlike vector space model metrics, text alignment algorithms compare two strings of characters by calculating the number of operations (either on single characters or on words) to transform one string into the other. Since a dependency exists among the characters/words their order in the text is relevant. Depending on the number of operations several algorithms have been defined [10]. We focus on two, Levenshtein distance and Jaro-Winkler distance.

Jaro-Winkler distance is an extension of the Jaro distance metric which takes typical spelling deviations into account. This extension modifies the weights of poorly matching pairs that share a common prefix.

Levenshtein distance is a simple edit distance function which calculates the distance by simply counting the minimum number of operations needed to transform one string into the other.

3.3 *N-gram Overlapping*

A common language-independent algorithm used in different NLP tasks is character or word n-grams overlapping. An n-gram is a subsequence of n characters or words of a given sequence of text. For similarity detection, n-grams overlapping measures the number of shared words n-grams between two texts [3]. The similarity measure is calculated using any appropriate similarity metric, such as Dice or Euclidian. The simplest way is by dividing the number of similar n-grams by the maximum number of n-grams [11].

A variation is the k-skip-n-grams overlapping that uses an n-grams distance metric, but takes into account a skip of k characters or words. Therefore, the characters or words need not be consecutive, there may be gaps in between [12]. This kind of n-grams allows to obtain nonconsecutive textual segments. The Rouge-n formula between two documents is:

$$\text{Rouge-}n = \frac{|n\text{-}grams \in \text{Can} \bigcap \text{Ref}|}{|n\text{-}grams \in \text{Ref}|}.$$

Can are the n-grams corresponding to the first document and Ref corresponds to the n-grams of the second document.

4 BUILDING THE CORPUS

The general purpose of our corpus is to automatically assess the similarity between a pair of texts. Therefore we evaluate different similarity

measures, either on the document or on the sentence level. That is, we try not only to find out whether two documents are similar, but also which sentences match.

Our corpus contains paraphrases of different complexity levels, from basic (for example by using synonyms) up to more complex ones. The difference to existing corpora lies in the granularity, in the ranking of paraphrase and in the annotation method for evaluation purposes. Our granularity is phrase-to-phrase. Furthermore every phrase is (to a different degree) modified as compared to the source phrase. So, the whole document is paraphrased. Related to the ranking, for the source text we obtain several levels of paraphrase. The first one relates to the bottom level, the second to an upper level and progressively up to the top level. Finally, we annotate the phrases of the source text mapping with the paraphrased document.

4.1 *Subject and Structure of the Corpus*

At the beginning it was decided to base the experiment on an article in German on Wikipedia[9] and build the corpus around the subject of the article. To limit the amount of paraphrasing work the article should consist of approximately 30 phrases. A particular cake (*Baumkuchen*[10]), very well known in Germany, was chosen as the subject of the study. As the original article contained slightly more phrases than required, a small number (less than 10) of phrases were deleted to obtain the version (31 phrases) used in our experiment.

To achieve the goals of the experiment, the corpus was partitioned into three subsets of documents, all to be evaluated for their similarity to the Wikipedia document. By systematically applying the rules explained below to the original article, two sets were obtained, each containing 5 manually rewritten (modified) documents. These sub-corpora are called *basic* and *complex paraphrase*.

The third sub-corpus was constituted for control purposes. It consists of 10 documents found on the WWW by a careful manual search. All documents were selected after thorough evaluation. The author read them all to make sure that the subject (the cake) was adequately addressed in the article. The documents of this control corpus were further divided into

[9] `http://de.wikipedia.org/wiki/Baumkuchen`

[10] For the English version see also `http://en.wikipedia.org/wiki/Baumkuchen`

two categories, of 5 documents each. The first category comprises minimally modified versions of documents that had been cited in the original Wikipedia article. Consequently, the degree of similarity to the original should be very high.

None of the documents of the second category had been cited by Wikipedia, i.e. their similarity to the original ought to be much lower. Those documents were also found on the web, but they treat the subject (the cake) from (completely) different angles as compared to the original article, i.e. an interview with the general manager of the company that makes and sells the cake in Japan, where it has made its very successful entry in the 1950s, an article from a German women's magazine, one that proposes a recipe that does without eggs, etc.

4.2 *Rules Concerning Form and Structure of Paraphrased Documents*

The manual rewriting process to obtain paraphrased versions of the original article was guided by a set of rules mainly specifying the permitted alterations to the structure and syntax of the source article. However, these rules had to be applied sensitively such that the narrative of the resulting article remained cohesive and comprehensible for a human reader.

Basic paraphrase almost ruled out a change of the length of an article, i.e. no more than one phrase was to be added to or deleted from the original. Equally, exchanges of segments (sub-phrases) among different phrases of the original article were forbidden. However, segments within a phrase might be arranged in a different sequence (intra-phrase), including the elimination of sub-phrases. The order of phrases in the paraphrased version of the article might also be a permutation of the original article. As the article focused on four main aspects of the cake, i.e. history, recipe, production process and geographical reach, the number of semantically acceptable permutations was rather limited by the requirement that the paraphrased version had to be comprehensive and cohesive.

Complex paraphrase gave more leeway to the rewriting process by permitting the insertion of up to 5 new phrases into the document plus the deletion of up to 5 phrases. In addition, exchanging segments between (several) phrases of the original article was allowed and encouraged. That is, complex paraphrase both makes use of inter-phrase and intra-phrase exchange of segments (sub-phrases). The term exchange was defined in a very general way. It encompasses splitting one phrase into two phrases or merging two phrases into one.

Another rule stipulated that none of the phrases of the original document was allowed to appear unaltered in any of the manually paraphrased versions. Yet small changes to the phrase (in the source document) to be paraphrased, like the removal of an adjective or an element within the enumeration of several alternatives, were sufficient to make the paraphrased document comply with that particular rule.

4.3 *Paraphrasing the Meaning of the Text*

There were no "semantic" rules attached to modifications of the original document, as long as the meaning (narrative/content) of the resulting document was more or less equivalent to the original. This allowed for using more general or more specific terms, the omission of details, the use of synonyms, different representations of information, etc. Several known standard techniques and tricks were applied and novel ideas developed as the author became more sophisticated in the process of generating further variants of paraphrased documents. Documents edited at a later time typically made use of knowledge acquired in all previous steps, unless the level of sophistication was deliberately reset or degraded.

The following paragraphs give an overview of all the techniques used in the experiment. The complex ones are generally found in documents of the complex paraphrase corpus. Also note that the following examples are represented in English, with an as faithful translation as possible. Among the simpler techniques, a few well known modifications that work for most of the languages shall be mentioned:

- Abbreviations: "vs" or "versus".
- Numbers: can be written as a sequence of ciphers (15) or as text (fifteen), small numbers also in Roman style (XV).
- Enumerations (reordering and suppression): "nutmeg, cinnamon and cardamom" vs. "cardamom and nutmeg".
- Hyperonyms and hyponyms: "sugary substance" vs. "honey" vs. "bee honey", "wood" vs. "pinewood".
- Synonyms and definitions: "manuscript" is "handwritten document".

More sophisticated modifications, some due to the intricacy of the German language, were:

- Compound words: a feature of the German language is the extensive use of compound words of very often considerable length. Where in

English "production of cake" is the proper term, in German "Produktion von Kuchen" or "Kuchenproduktion" are synonymous, with the latter one stylistically preferable. Further elaborating on the example "Kuchenproduktionsverfahren" in English requires at least twice the separator "of". In German there are several rewritings, which all may be used as a paraphrase.

– Complex phrase structure: another feature of the German language is a certain tendency to formulate lengthy phrases of complex structure, i.e. containing (several layers of nested) several sub-phrases. There typically are many simpler rewritings or the possibility to reorder those sub-phrases without changing the meaning.

Complex "semantically" paraphrase was achieved by generalizing temporal and geographical entities, indirect definitions of persons that cannot be deduced from other phrases of the original text. Approximations of quantities have also been used regularly:

– Temporal: 1855 vs. "in the midst of the nineteenth century" vs. "between 1840 and eighteen hundred and sixty-two".
– Geographical: "Dresden" vs. "the capital of Saxony", "Japan" vs. "the land of the rising sun", "Masuria" vs. "north-eastern region of Poland".
– Personal: "Prince Elector Frederick William" vs. "the Head of State of Brandenburg" or "Karl Joseph Wilhelm Juchheim" vs. "German patissier".
– Quantitative: 8 vs. "between 6 and 9" vs. "a one digit number".

As the paraphrased documents became ever more sophisticated, several techniques were applied to the same text passages, for example synonym and generalization.

4.4 *The Basic/simple Paraphrase Sub-corpus*

To start with, the five documents of the basic paraphrase corpus were created. The first document was obtained by applying some of the rather basic changes mentioned above to the original Wikipedia article. Subsequent documents were built on the text of documents that had been created in a previous step. Sometimes a new phrase was added or deleted in accordance with the structural rules. Then further changes were made, such that step by step the list of techniques mentioned above was elaborated. To check for the effect of permutations at least one of the paraphrased versions was a simple permutation of a previous one, may be

plus one additional phrase or minus one deleted phrase. It is worthwhile to mention that the last document of the basic paraphrase corpus made use of almost all of the techniques that had been developed, but it respected all the structural rules associated with the basic corpus, in particular no exchange of segments and minimal or no change in the length of the document.

4.5 *The Complex Paraphrase Sub-corpus*

The first document of the complex paraphrase corpus was created in order to evaluate the effect of bigger structural changes. Therefore several phrases were added to and deleted from a document of the basic paraphrase corpus, one phrase was merged and one split. The sequence of phrases was not changed.

The second document was based on the first one, with more splits, merges and exchanges of segments between phrases. Furthermore the order of the phrases was permuted. The level of paraphrase was also raised. Among others, all numbers were written in textual form or vice versa, geographic and temporal references were generalized and the use of synonyms and definitions of terms increased.

The third document used one of the more sophisticated documents of the basic paraphrase corpus as a starting point. The level of paraphrase was increased by using techniques from the list above more frequently. Phrases were added, removed, split and merged, but no segments moved between phrases.

The fourth document took the most sophisticated document of the basic paraphrase corpus as a starting point. Special effort was then made to apply all the techniques mentioned so far to the maximum, especially generalization of personal, geographic and other entities. New paraphrases were devised that had not been used in other documents of either of the sub-corpora. The maximum allowed number of phrases was added and deleted, several phrases merged and split. In addition, several segments were moved between phrases. Finally, the phrases were reordered in compliance with the comprehensibility requirement. In essence, the fourth document is the most elaborately paraphrased of all the documents in the entire corpus.

As almost all the options had been applied to a certain degree, the fifth document was compiled by selecting sets of phrases from several previously mentioned documents, then deleting the maximum allowed num-

ber, adding 3 new phrases, merging some and moving segments where appropriate.

5 Evaluation Aids

Similarity between the original document and each of the paraphrased versions was calculated by applying the techniques and formulas described in the previous sections. Basically, the algorithms tried to determine the most similar phrases in the original to any of the phrases in the paraphrased document and vice versa. To be able to assess the precision of those algorithms' predictions, a simple method to specify the relationship between the original and the paraphrased document was devised.

The mapping between phrases of the original document and the phrase or phrases in the paraphrased documents was noted, transformed into a predefined format and then stored into a small file specific to each document. This file would contain one line for each phrase in the original. The line begins with the number of the phrase in the original document, then a colon and then the number of the phrase in the paraphrased document to which the original was rewritten. The following examples illustrate this mapping technique. Please also note that "phrase X of the original is found in phrase Y of the paraphrased document" is shorthand for "all or part of the meaning of phrase X after much paraphrasing can be found in phrase Y".

- "0:29" indicates that phrase 0 of the original is found in phrase 29 of the paraphrased document.
- "2:" indicates that phrase 2 of the original has been removed.
- "3:1" together with "4:1" indicate that phrases 3 and 4 from the original have been merged into phrase 1 of the paraphrased document.
- "13:11,12" indicates that phrase 13 from the original has been split and may be found in phrases 11 and 12 of the paraphrased document.
- "5:5,6" together with "6:5,6" indicate that segment of phrases 5 and 6 from the original have been exchanged.
- If there is a phrase in the paraphrased document the number of which is to be found nowhere to the right of the colon in the file specifying the mapping, that means this phrase has been added to the original.

6 Use of our Corpus

We assessed the average values of the simple similarity measures calculated on the 15 texts of the German corpus (five texts of the basic para-

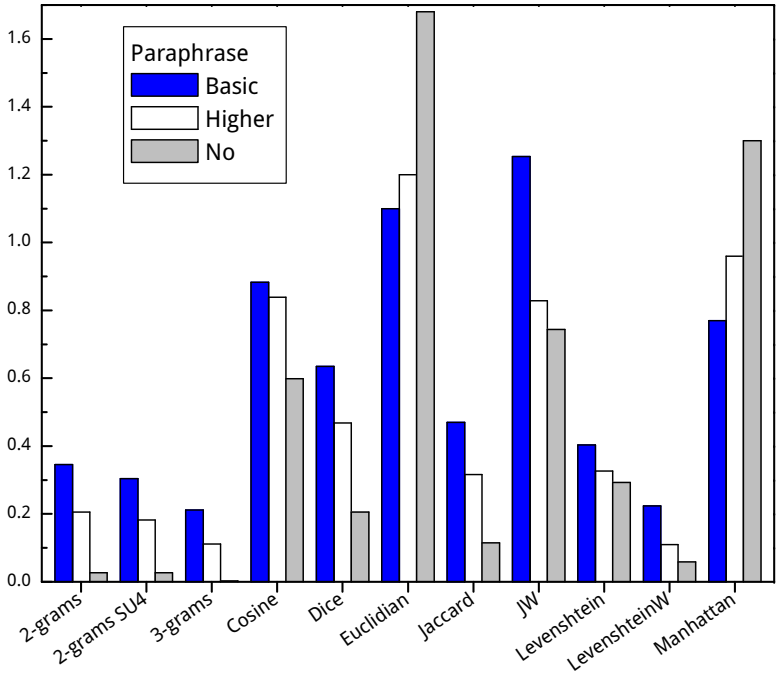


Fig. 1. Text similarity measures calculated over the German corpus: not paraphrase

phrase sub-corpus, five texts of the complex paraphrase sub-corpus and five texts not related with the source text). Except for Euclidean distance, Manhattan distance and Jaro-Winkler (JW), the measures were normalized to the range $[0, 1]$.

Values close to 0 indicate high proximity between the source text and the suspicious text (although for the Euclidean, Manhattan and JW distance higher values indicate more difference between both texts).

As can be seen in Figure 1, the highest similarity scores correspond to paraphrases of the basic level, following the higher level and finally the lowest score for no paraphrases (but inverted scores for Euclidean and Manhattan distances).

We also calculated the Pearson correlation factor [13] among all the similarity measures to assess the score among overall simple measures. Table 1 shows a strong correlation as was expected.

Table 1. Correlation of similarity measures

Similarity	Low	High	Not	Pearson
2-grams	0.34565	0.20559	0.02875	0.97716
2-grams SU4	0.30396	0.18248	0.02971	0.97692
3-grams	0.21197	0.11161	0.00178	0.96765
Cosine	0.88245	0.83879	0.63233	0.99693
Dice	0.63473	0.46795	0.22470	0.98494
Euclidean	1.07344	1.22539	1.67582	–1
Jaccard	0.46983	0.31623	0.12709	0.97562
JW	1.25345	0.82801	0.73783	0.80201
Levenshtein	0.40322	0.32613	0.28818	0.88927
Levenshtein W	0.22375	0.10933	0.05669	0.88169
Manhattan	0.78558	0.95836	1.29115	–0.995

7 CONCLUSIONS

We presented a new German corpus for paraphrasing detection. It features two particular characteristics as compared to current corpora, i.e. a set of aligned text pairs at paraphrase level with a file of references, and three levels of paraphrase for each document (high, low and no paraphrase).

The mapping between the source text and the paraphrased one lets even persons that do not speak German study the corpus for similarity detection, thereby making it language independent. Furthermore, the level of paraphrase allows to refine the algorithms for paraphrasing detection, in order to determine the degree of paraphrase more precisely.

The application of the simple measures on our corpus and of the Pearson correlation factor overall measures let us see, as expected, how the similarity score increases in direct proportion to the degree of paraphrase.

We are still incorporating new texts into the German corpus. Besides, we are preparing two other corpora, a Spanish and a French corpus, using the same protocol as the one presented in this paper. Meanwhile, the German corpus is available online at the website: `http://simtex.talne.eu`.

ACKNOWLEDGMENTS We acknowledge Mexican *Consejo Nacional de Ciencia y Tecnología* (CONACYT) grant number 178248 and Project UNAM-DGAPA-PAPIIT number IN400312.

REFERENCES

1. Goldstone, R.L.: The role of similarity in categorization: Providing a groundwork. Cognition **52**(2) (1994) 125–157
2. Bendersky, M., Croft, W.B.: Finding text reuse on the web. In: Second ACM International Conference on Web Search and Data Mining (WSDM'09), New York, NY, USA, ACM (2009) 262–271
3. Clough, P., Gaizauskas, R., Piao, S.S.L., Wilks, Y.: METER: MEasuring TExt Reuse. In: 40th Annual Meeting on Association for Computational Linguistics (ACL'02), Stroudsburg, PA, USA, ACL (2002) 152–159
4. Gaizauskas, R., Foster, J., Wilks, Y., Arundel, J., Clough, P., Piao, S.: The Meter corpus: A corpus for analysing journalistic text reuse. In: Conference on Corpus Linguistics 2001. (2001) 214–223
5. Dolan, W.B., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In: 20th International Conference on Computational Linguistics (Coling'04), Stroudsburg, PA, USA, ACL (2004) 350–356
6. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: 3rd International Workshop on Paraphrasing (IWP'05). (2005) 9–16
7. Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In: PAN'09, Stein, Rosso, Stamatatos, Koppel, Agirre (Eds.) (2009) 1–9
8. Lin, D.: An information-theoretic definition of similarity. In: Fifteenth International Conference on Machine Learning (ICML'98), San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1998) 296–304
9. Bär, D., Biemann, C., Gurevych, I., Zesch, T.: UKP: Computing semantic textual similarity by combining multiple content similarity measures. In: First Joint Conference on Lexical and Computational Semantics (SemEval'12). Volume 1., Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 435–440
10. Ali, A.: Textual similarity. BSc thesis, Technical University of Denmark, DTU Informatics, Lyngby, Denmark (2011)
11. Gomaa, W.H., Fahmy, A.A.: A survey of text similarity approaches. International Journal of Computer Applications **68**(13) (2013) 13–18
12. Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y.: A closer look at skip-gram modelling. In: Fifth international Conference on Language Resources and Evaluation (LREC'06). (2006)
13. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: American Association for Artificial Intelligence (AAAI'06). (2006) 775–780

JUAN-MANUEL TORRES-MORENO
LABORATOIRE INFORMATIQUE D'AVIGNON,
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE,
FRANCE
AND
DEPARTEMENT DE GÉNIE INFORMATIQUE,
ÉCOLE POLYTECHNIQUE DE MONTRÉAL,
CANADA
E-MAIL: <JUAN-MANUEL.TORRES@UNIV-AVIGNON.FR>

GERARDO SIERRA
LABORATOIRE INFORMATIQUE D'AVIGNON,
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE,
FRANCE
AND
INSTITUTO DE INGENIERÍA,
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO,
MEXICO
E-MAIL: <GSIERRAM@IINGEN.UNAM.MX>

PETER PEINL
FACULTY OF APPLIED INFORMATICS,
UNIVERSITY OF APPLIED SCIENCES FULDA,
GERMANY
E-MAIL: <PETER.PEINL@INFORMATIK.HS-FULDA.DE>

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 25–47
Received 17/02/2014, accepted 23/04/2014, final 2/07/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Should Syntactic N-grams Contain Names of Syntactic Relations?

GRIGORI SIDOROV

Instituto Politécnico Nacional, Mexico

ABSTRACT

In this paper, we discuss a specific type of mixed syntactic n-grams: syntactic n-grams with relation names, snr-grams. This type of syntactic n-grams combines lexical elements of the sentence with the syntactic data, but it keeps the properties of traditional n-grams and syntactic n-grams. We discuss two possibilities related to labelling of the relation names for snr-grams: based on dependencies and based on constituencies. Examples of various types of n-grams, sn-grams, and snr-grams are given.

1 INTRODUCTION

In our previous works starting in 2012 we proposed a concept of syntactic n-grams (Sidorov, Velasquez, Stamatatos, Gelbukh & Chanona-Hernandez 2012, 2013, 2014; Sidorov 2013a, 2013b, 2013c). This concept is quite on the agenda of the computational linguistics: say, our works obtained many positive feedback comments, besides, the same concept was implemented independently for English language in the form of a large collection of syntactic n-grams obtained from books by (Goldberg and Orwant, 2013), while they were working on this project in Google.

Let us remind that syntactic n-grams are n-grams of textual elements obtained in a specific non-linear manner based on syntactic relations (Sidorov 2013c), i.e., instead of using the order of elements in the surface

structure, the syntactic structure is used. For obtaining syntactic n-grams, we traverse the syntactic tree and use the order of elements in it. It is equivalent (but probably less clear) to say that we use subtrees of a syntactic tree as syntactic n-grams. It is obvious that the syntactic structure is non-linear with respect to the surface structure: the order of elements is usually changed. We discuss the concept of syntactic n-grams in greater detail in the next section.

Syntactic n-grams can be used in any task in the field of the Natural Language Processing, when traditional n-grams can be applied. It is especially important in the modern paradigm related to application of machine learning algorithms, because this paradigm is completely based on the concept of vector space model and feature selection, where the features are precisely n-grams or syntactic n-grams.

Syntactic n-grams are similar in nature to so-called concepts (Poria, Agarwal, Gelbukh, Hussain & Howard 2014). Use of concepts in sentiment analysis has become very popular and set up a new research field called concept-level sentiment analysis (Poria, Cambria, Ku, Gui & Gelbukh 2014; Poria, Ofek, Gelbukh, Hussain & Rocach 2014). In standard sentiment lexicons concepts are usually ignored. However, modern research shows that concepts carry meaning and sentiment, and they are more useful for, for example, sentiment analysis than word-level approaches (Cambria, Poria, Gelbukh & Kwok 2014). Concepts are also useful to understand emotions (Das, Poria & Bandyopadhyay 2012). For this and other tasks, concept vectors are used instead of bag of words (Cambria, Fu, Bisio & Poria 2015; Poria, Gelbukh, Cambria, Hussain & Huang 2014).

Machine learning simulates human ability for classification of objects based on their similarity. The best features for similarity calculation depend on a specific task. For example, for thematic classification of documents we need to take into account words thematically related to each topic and ignore auxiliary words, while, say, for analysis of author's writing style we would prefer to focus precisely on auxiliary words, because they may reflect the style. Both supervised and unsupervised machine learning algorithms can be applied using syntactic n-grams as features in the corresponding vector space model.

An alternative to machine learning methods is the paradigm based on formulation and application of handcrafted rules. This paradigm was prevalent until the end of the 20th century (Bolshakov, Gelbukh 2004). In this paradigm, the human evaluators analyze the example data of the

problem, try to propose some hypotheses about the structure and function of the phenomena related to the problem and after this, extract problem-dependent features, and formulate rules. These rules usually correspond to selectional preferences, i.e., the generalized restrictions on combination between elements. The current state of the art is that machine learning algorithms—if they have sufficiently large marked corpus for training—outperform human crafted rules. Note that the human effort is still present, though it is moved from formulation of the rules to marking of the corpora (Gelbukh 2013).

The advantage of machine learning algorithms over humans is that these algorithms are consistent and consider many variants during feature selection using vast data, while humans are not consistent, cannot process big volumes of data, and cannot generalize over too many examples. Obviously, the humans are better than the computers while marking the corpora using intuition, because they can use the extra linguistic world knowledge and common sense, which computers do not possess, for understanding of individual sentences or texts. However, it seems that given a marked corpus, a machine-learning algorithm can perform better feature selection than a human can.

The paper is organized as follows. Dependency and constituency representations of syntactic relations are discussed in Section 2. In Section 3, we describe the concept of syntactic n-grams and present their various types. In Section 4 we propose the concept of syntactic n-grams with relation names (snr-grams) and give some examples of their extraction using formalisms of dependencies and constituencies. Finally, conclusions are drawn in Section 5.

2 Constituencies vs. Dependencies as Syntactic Representations

There are two main formalisms for representation of syntactic structure: dependencies and constituents. The dependency formalism directly reflects relations between words, usually using arrows. Since one word in a syntactic relation is the headword, while the other one is the dependent word, the arrow has the direction: head → dependent. The arrows are labelled with the types of syntactic relations. If there is no natural head, like, say, in case of a coordinative relation, some decision about the head/dependent words should be made anyway.

The constituency formalism represents syntactic relations with respect to the underlying formal grammar and reflects the history of the syntactic tree derivation according to this grammar. The syntactic relations between words are established based on the applied grammar rules: derivation history. Note that some relations are established not between words themselves, but between constituents, which represent the result of the previous application of the rules.

Constituency trees have longer history in usage in the computational linguistics, because they are directly related to application of generative grammars (N. Chomsky). Modern approaches pay more attention to dependency trees, because they are more natural and direct. Besides, they contain the information about the syntactic roles of words, like "direct object", "subject", etc.

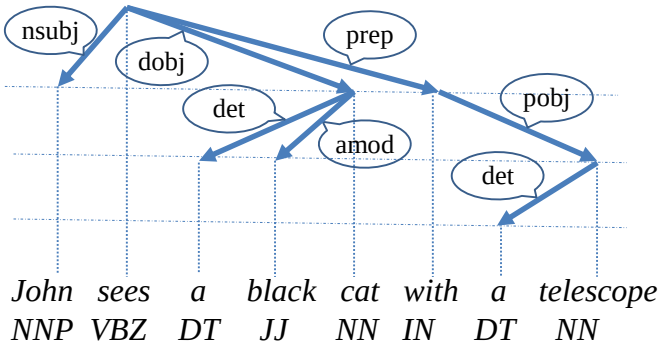


Fig. 1. Example of a dependency tree

2.1 *Example of the Representation of a Syntactic Tree*

Let us present an example of the dependency and constituency formalisms for a syntactic tree, for instance, for the phrase *John sees a black cat with a telescope*. The syntactic tree that uses dependency formalism is shown in Fig. 1. We also show the POS tags of each word on the next line below the corresponding word.

The example of representation of the same phrase using the formalism of constituencies is shown in Fig. 2. In this case, we mark with wider line the part of the constituent that corresponds to the headword.

We also show in the tree structure the left parts of the applied rules, i.e., the generalization introduced by each rule.

This constituency tree is generated by the following very simple formal grammar. It is clear that real parsers can use more complex or more general rules, but for our discussion, this grammar is sufficient. We mark with "*" the head elements in the rules:

S → NNP VP*
VP → VP* PP
NP → JJ NN*
NP → DT NN*
NP → DT NP*
PP → IN* NP
VP → VBZ* NP

The derivation history of the phrase is the order of application of the grammar rules. For example, we start with the rules that correspond directly to words (terminal nodes) "NP → DT NN*", "NP → JJ NN*". After this, the "intermediate" rules like "VP → VBZ* NP" are applied and finally the "top" rule "S → NNP VP*" is used. This derivation history corresponds to the analysis strategy "bottom-up", being the other possible strategy the reverse order of application of the rules: "top-down".

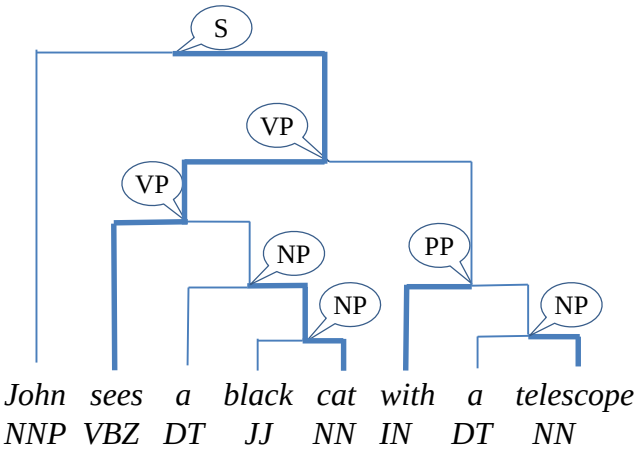


Fig. 2. Example of a constituency tree

2.2 *Conversion between Constituencies and Dependencies*

It is well-known that dependency and constituency formalisms are equivalent in general, i.e., there exists an algorithm that transforms the dependency tree structure into the constituency tree structure and vice versa (Gelbukh, Calvo, Torres 2005). It is not surprising, because both types of trees reflect the same syntactic reality. Note that this is only general (structural) conversion, as it does not convert the syntactic labels in both directions.

The algorithm for constituency to dependency general conversion is simple. For each word that is a head word (it is marked with "wider" line) go up in the tree. At each step (while going up following the constituents) go down to a dependent constituent. After this follow downwards the head relations only (the "wider" line) and draw the arrow from the headword to the obtained dependent word. Continue going up in the tree from the point, when you start going down.

For constituency to dependency general conversion, the formal grammar should mark the words that are heads on the right side of the rules, because otherwise we would not know the directions of the dependency arrows. Note that if the grammar does not mark them, the marking can be done in a random manner, but obviously with not so good results: the conversion will be done, but some arrows would have anti-intuitive directions. It is also clear that the resulting dependency tree does not contain the names of syntactic relations for the arrows.

The algorithm for dependency to constituency general conversion is also simple. We start with arrows at the lowest level and go to upper levels. For each arrow, we establish a constituent relation for the pair of words, being the headword the starting point of the arrow. If the headword already forms a constituent, then this constituent should be used instead of the word itself. Some additional conventions are necessary, for example, in case of bifurcations, we first process the arrows that are the closest ones to the word, or that *nsubj* relation is processed last.

It is clear that for dependency to constituency general conversion the resulting constituency tree does not have the interpretation of constituents (left parts of the rules represented in the tree structure), because it is precisely what the formal grammar does; in certain sense, the resulting representation will lack of generalization for constituents.

3 SYNTACTIC N-GRAMS

As we mentioned above, we introduced the concept of syntactic n-grams in our previous works (Sidorov et al. 2012, 2013, 2014; Sidorov 2013a, 2013b, 2013c). Similar ideas were proposed in (Pado, Lapata 2007; Gelbukh 1999), but they were treated as very specific methods for certain tasks of syntactic or semantic analysis. The importance of the concept is confirmed by the fact that Google obtained and made public syntactic n-grams for a large set of books in English (Goldberg and Orwant, 2013).

In our earlier works, we preferred to use the term "syntactic dependency-based n-grams", adding the words "dependency-based". It was important, because there is possible naive misinterpretation of the term "syntactic n-grams" as "sequence of POS tags", because POS tags are perceived as carrying some syntactic information. In fact, it is not true: POS tags are more morphological than syntactic phenomena—the syntactic information is used only for disambiguation between several possible POS tags for a word. At most, we can consider them as morphosyntactic entities. Now, as the term "syntactic n-grams, sn-grams" is more habitual, we can omit the words "dependency based". Note that we say "dependency based" (and not "constituency based"), because syntactic dependencies are much more direct projection of syntactic paths for construction of sn-grams. Constituencies can be applied to construction of sn-grams as well, though not so naturally, see discussion in Sections 2 and 4.

So, while traditional n-grams are sequences of textual elements (words, POS tags, etc.) taken as they appear in texts, the general idea behind syntactic n-grams is to take the surface textual elements in a non-linear order by following paths in syntactic trees. In this case, the order of textual elements is usually changed in comparison with the surface structure.

3.1 *Types of Syntactic N-grams*

In our previous works, we have proposed the classification of syntactic n-gram types. Depending on the elements that constitute them, there can be syntactic n-grams of words/lemmas/stems (lexical elements), POS tags, SR tags (names of Syntactic Relations), multiword expressions (Gelbukh, Kolesnikova 2013a, 2013b; Ledeneva, Gelbukh, García-Hernández 2008), and even of characters (Sidorov et al. 2013, 2014).

For obtaining character sn-grams, we first construct sn-gram of lexical units (words or lemmas) and then character sn-grams are constructed over this sequence in the same way as for traditional character n-grams. Note that for this procedure it is preferable to use sn-grams that contain most number of elements (long sn-grams, n-grams with large values of n), but each lexical element should be considered at least once. There is a problem for future research: how to calculate correctly the frequencies of character sn-grams obtained in this manner, because many elements in sn-grams are repeated.

There also can be mixed sn-grams, e.g., one element in an sn-gram is a POS tag and the other one is a lexical unit. Note that character sn-grams cannot be naturally mixed with other types of sn-grams, because they have different nature: all other types of sn-grams reflect properties of words (lexical unit, POS tag), even SR tags reflect the relations of a word with other word, while character sn-grams are sequences of characters obtained from already existing sn-grams of lexical units, so they are derivate, and in a certain sense they are secondary. We insist on considering them because in certain tasks, like, for example, authorship attribution, traditional character n-grams quite surprisingly give very good results, so character sn-grams should be tried as well.

On the other hand, in (Sidorov, 2013a) we have proposed differentiating between continuous (non-interrupted path, path without bifurcations) and non-continuous (path with interruptions or returns (or bifurcations)) syntactic n-grams. It is obvious that continuous sn-grams are a special type of non-continuous sn-grams, namely sn-grams with no returns (without bifurcations). Intuitively, we consider that continuous sn-grams can contain more important linguistic information, but it should be verified for various tasks in the experimental manner. It is clear that for syntactic bigrams there is no difference between continuous and non-continuous sn-grams, because no bifurcations are possible in case of exactly two elements in an n-gram.

Note that here appears another possible naïve misinterpretation of the general term "syntactic n-grams" that would be "n-grams of names of syntactic relations (SR tags)". It is possible, since these sn-grams can be obtained only if we apply parsing before. Nevertheless, this interpretation is too narrow: yes, it is the possible type of sn-grams, but there are other types as well. In general, when speaking about syntactic n-grams we refer not to a specific type of elements (SR tags, words, etc.) but to the manner of their construction by following paths in syntactic trees.

Another important consideration is related to the traditional practice of treatment of stop words (auxiliary words). There are two possibilities: taking them into account vs. filtering out of stop words. The possibility of filtering out of stop words can be easily applied to syntactic n-grams: we should follow syntactic paths and when we encounter with a stop word, we ignore it and just continue with the next word according to the path. In fact, this idea was generalized as "filtered n-grams" in (Sidorov 2013c): we can filter out not only stop words, but any words that do not comply with any chosen criterion, for example, it can be a thresholds based on tf-idf values.

Finally, we would like to mention that the elements of the same level in an sn-gram can be taken as they appear in the sentence, or can be reordered according to some criteria, for example, using the alphabetic order of the elements. The first possibility takes into account the word order in the sentences, while the second one tries to ignore possible (insignificant) changes in the word order.

3.2 *Extraction of Syntactic N-grams and their Representation*

The software for extraction of syntactic n-grams is available on the Web page of the author[1]. It takes as the input the file generated by the Stanford parser (de Marneffe, MacCartney, Manning 2006) and it produces sn-grams of the desired size and type.

Note that the software also treats in a practical manner the problem of exponential growth of the number of sn-grams in case of too many dependents of a word. This problem consists in the fact that if the number of the dependent words is large, say, more than six, then the number of possible combinations (i.e., non-continuous sn-grams) may become too large. It is very rare situation to have so many dependent words, but it may appear in real life, especially if something went wrong with parsing or if we want to treat punctuation (like parenthesis) and the parser chooses one word as their head.

For the example in Fig. 1 and Fig. 2, the Stanford parser generates the output presented in Fig. 3 and Fig. 4 correspondingly.

Let us now discuss how to represent syntactic n-grams. If we use only continuous sn-grams, then we can represent them using the sequences of words just like in case of the traditional n-grams. But if we

[1] http://www.cic.ipn.mx/~sidorov

```
nsubj(sees-2, John-1)
root(ROOT-0, sees-2)
det(cat-5, a-3)
amod(cat-5, black-4)
dobj(sees-2, cat-5)
prep(sees-2, with-6)
det(telescope-8, a-7)
pobj(with-6, telescope-8)
```

Fig. 3. Results of the analysis using dependencies

```
(ROOT
 (S
  (NP (NNP John))
  (VP (VBZ sees)
   (NP (DT a) (JJ black) (NN cat))
   (PP (IN with)
    (NP (DT a) (NN telescope))))
  ))
```

Fig. 4. Results of the analysis using constituencies

start considering non-continuous sn-grams, then it turns out that we need special metalanguage for their representation, namely for distinguishing the words that form a sequence from the words that have returns in the path (bifurcations).

For example, if we have three words A, B, C and we want to express that both B and C are dependent from A, i.e., there is a return in the path (a bifurcation), then we separate B and C with a comma and put them into the brackets: "A [B, C]". If there is no bifurcation that means that C depends from B and B depends from A, then we just write "A B C". In the current version of our software we add more brackets: "[A[B[C]]]" and "[A[B,C]]". This notation reflects more consistent use of brackets in each node and better shows the underlying tree structure. Note that if used uniformly, it does not affect the identity of sn-grams.

3.3 *Example of Extraction of Syntactic N-grams*

Let us consider the example presented in Fig. 1. The second line of the example contains the POS tags for each word. It is obvious that we can

substitute lexical units with their lemmas, for example, use *see* instead of *sees*, as well as with their POS tags, for instance, use VBZ instead of *sees* or NN instead of *telescope*, etc. Thus, we suppose that the reader understands this possibility and we will not illustrate it in the figure and in further discussion: we should just remember that while we use words, they can as well be substituted by lemmas or POS tags, or any combinations of these elements. As we mentioned in our previous works, it is a question of future experimental research to determine what types of sn-grams or mixed sn-grams are useful for particular tasks.

Table 1. Traditional and syntactic bigrams

Traditional bigrams	Syntactic bigrams
John sees	*sees[with]*
sees a	*telescope[a]*
a black	*sees[cat]*
black cat	*cat[black]*
cat with	*with[telescope]*
with a	*cat[a]*
a telescope	*sees[John]*

Table 2. Traditional and syntactic trigrams

Traditional trigrams	Syntactic trigrams
John sees a	*with[telescope[a]]*
sees a black	*sees[cat,with]*
a black cat	*sees[John,cat]*
black cat with	*sees[John,with]*
cat with a	*sees[cat[a]]*
with a telescope	*sees[with[telescope]]*
	cat[a,black]
	sees[cat[black]]

Table 3. Traditional and syntactic 4-grams

Traditional 4-grams	Syntactic 4-grams
John sees a black	*sees[cat[a,black]]*
sees a black cat	*sees[John,with[telescope]]*
a black cat with	*sees[cat[a],with]*

black cat with a	*sees[John,cat[black]]*
cat with a telescope	*sees[John,cat,with]*
	sees[cat,with[telescope]]
	sees[John,cat[a]]
	sees[cat[black],with]
	sees[with[telescope[a]]]

Table 4. Traditional and syntactic 5-grams

Traditional 5-grams	Syntactic 5-grams
John sees a black cat	*sees[John,cat[black],with]*
sees a black cat with	*sees[cat[a,black],with]*
a black cat with a	*sees[John,with[telescope[a]]]*
black cat with a telescope	*sees[John,cat[a,black]]*
	sees[cat[black],with[telescope]]
	sees[cat[a],with[telescope]]
	sees[John,cat,with[telescope]]
	sees[cat,with[telescope[a]]]
	sees[John,cat[a],with]

Table 5. Traditional and syntactic 6-grams

Traditional 6-grams	Syntactic 6-grams
John sees a black cat with	*sees[John,cat[a,black],with]*
sees a black cat with a	*sees[John,cat[a],with[telescope]]*
a black cat with a telescope	*sees[John,cat,with[telescope[a]]]*
	sees[cat[black],with[telescope[a]]]
	sees[cat[a],with[telescope[a]]]
	sees[John,cat[black],with[telescope]]
	sees[cat[a,black],with[telescope]]

Table 6. Traditional and syntactic 7-grams

Traditional 7-grams	Syntactic 7-grams
John sees a black cat with a	*sees[John,cat[a],with[telescope[a]]]*
sees a black cat with a telescope	*sees[cat[a,black],with[telescope[a]]]*
	sees[John,cat[black],with[telescope[a]]]
	sees[John,cat[a,black],with[telescope]]

Let us extract all possible traditional n-grams and syntactic n-grams of various sizes and types from the example sentence. First, we present traditional n-grams of words of various sizes and syntactic n-grams of the same sizes in Tables 1-6. We start with bigrams and go till 7-grams. Note that in practical tasks of the computational linguistics, we usually do not need larger size of n-grams, because they do not repeat any more in texts, i.e., their frequency is always equal to 1 in any corpus and they are practically useless.

It can be observed that syntactic n-grams are much more linguistically motivated, because for their construction we use very important linguistic knowledge: syntactic structure. For example, traditional n-grams like "with a" or "sees a" no longer form part of the features for machine learning algorithms. A counterargument might be that these n-grams can appear consistently in the corpus. The answer to this counterargument is that though it is true, these n-grams contain more noise than real information, because there is no linguistic reality behind them.

Now let us present syntactic n-grams of SR tags, Tables 7–12. Obviously, there are no traditional n-grams using this type of elements.

Table 7. Syntactic bigrams of SR tags

Syntactic bigrams
prep[pobj]
root[nsubj]
root[prep]
root[dobj]
dobj[amod]
pobj[det]
dobj[det]

Table 8. Syntactic trigrams of SR tags

Syntactic trigrams
prep[pobj[det]]
root[dobj,prep]
root[dobj[amod]]
root[nsubj,dobj]
dobj[det,amod]
root[prep[pobj]]
root[nsubj,prep]
root[dobj[det]]

Table 9. Syntactic 4-grams of SR tags

Syntactic 4-grams
root[dobj,prep[pobj]]
root[nsubj,dobj,prep]
root[prep[pobj[det]]]

root[dobj[det],prep]
root[nsubj,dobj[amod]]
root[dobj[amod],prep]
root[nsubj,prep[pobj]]
root[nsubj,dobj[det]]
root[dobj[det,amod]]

Table 10. Syntactic 5-grams of SR tags

Syntactic 5-grams
root[dobj[amod],prep[pobj]]
root[dobj,prep[pobj[det]]]
root[nsubj,dobj[amod],prep]
root[dobj[det,amod],prep]
root[nsubj,dobj[det,amod]]
root[dobj[det],prep[pobj]]
root[nsubj,prep[pobj[det]]]
root[nsubj,dobj,prep[pobj]]
root[nsubj,dobj[det],prep]

Table 11. Syntactic 6-grams of SR tags

Syntactic 6-grams
root[nsubj,dobj[amod],prep[pobj]]
root[dobj[amod],prep[pobj[det]]]
root[dobj[det,amod],prep[pobj]]
root[nsubj,dobj[det,amod],prep]
root[dobj[det],prep[pobj[det]]]
root[nsubj,dobj[det],prep[pobj]]
root[nsubj,dobj,prep[pobj[det]]]

Table 12. Syntactic 7-grams of SR tags

Syntactic 7-grams
root[nsubj,dobj[det,amod],prep[pobj]]
root[nsubj,dobj[amod],prep[pobj[det]]]
root[dobj[det,amod],prep[pobj[det]]]
root[nsubj,dobj[det],prep[pobj[det]]]

In a similar manner, we can construct syntactic n-grams using constituency trees, namely, the derivation history, based on considerations presented in Section 4. We give the example of bigrams of relations based on derivation history in Table 13. The parentheses are used for containing the corresponding fragment of the derivation history. In this case, we consider that the relation "root" corresponds to the left part of the rule with the element "S". For comparison, we give also the syntactic bigrams based on SR tags from Table 7.

Table 13. Syntactic bigrams of derivation history fragments

Syntactic bigrams based on derivation history	Syntactic bigrams based on SR tags
(VP,VP,PP)[(PP,NP)]	*prep[pobj]*
(S)[(NN)]	*root[nsubj]*
(S)[(VP,VP,PP)]	*root[prep]*
(S)[(VP,NP,NP)]	*root[dobj]*
(VP,NP,NP)[(NP)]	*dobj[amod]*
(PP,NP)[(NP)]	*pobj[det]*
(VP,NP,NP)[(NP)]	*dobj[det]*

4 SYNTACTIC N-GRAMS WITH RELATION NAMES (SNR-GRAMS)

We hope that the reader now has clear idea about the concept of syntactic n-grams and their types. Among types of syntactic n-grams we mentioned that there can be mixed syntactic n-grams. In this sense, we already considered the fact that syntactic n-grams can contain names of syntactic relations mixed with other elements. Nevertheless, there are considerations for drawing attention to this particular type of sn-grams: they contain both lexical/morphological elements (words, lemmas, POS tags) and at the same time the names of syntactic relations (SR tags). Let us call these sn-grams that contain relation names "snr-grams", when we prefer to use the abbreviation.

It can be observed that snr-grams convey more information than any other type of n-grams or sn-grams, and still they can be used as features in machine learning tasks, when other n-grams can be used. So, we believe that this type of sn-grams deserves special attention. We have

preliminary information that snr-grams performed better in the task of the periphrasis as compared to n-grams and other types of sn-grams (personal communication of Hiram Calvo, to be published soon).

There is also a certain problem that consists in how to count the number of elements in snr-grams. If we count both words/POS tags together with SR tags then, say, there will be no bigrams, and in general, no n-grams with even values of n. So our suggestion, if we deal with snr-grams, is counting only the elements different from SR tags. In case that we deal with sn-grams of SR tags only, then, obviously, we should count these elements (SR tags). In general, if we want to use mixed n-grams, when certain elements are word based (words, POS tags) and the other elements are relation based (SR tags), we should count SR tags only if we do not want to take into account the word based elements. For example, if we want to consider syntactic bigrams, where the first element is the word and the second one is the SR tag, then we treat the SR tags as the proper element of the bigrams. On the other hand, if we are working with snr-grams, then SR tags should not be counted for determining the snr-gram size.

Tables 14–16 show snr-grams from the example above using SR tags as part of snr-grams. Parentheses before each word contain the relation name. Note that it should appear immediately before the word because of ambiguities of possible bifurcations.

Table 14. Snr-grams of size 2 (SR tags)

Snr-grams
sees[(prep)with]
telescope[(det)a]
sees[(dobj)cat]
cat[(amod)black]
with[(pobj)telescope]
cat[(det)a]
sees [(nsubj)John]

Table 15. Snr-grams of size 3 (SR tags)

Snr-grams
with[(pobj)telescope[(det)a]]
sees[(dobj)cat,(prep)with]

sees[(nsubj)John,(dobj)cat]
sees[(nsubj)John,(prep)with]
sees[(dobj)cat[(det)a]]
sees[(prep)with[(pobj)telescope]]
cat[(det)a,(amod)black]
sees[(dobj)cat[(mod)black]]

Table 16. Snr-grams of size 4 (SR tags)

Snr-grams
sees[(dobj)cat[(det)a,(amod)black]]
sees[(nsubj)John,(prep)with[(pobj)telescope]]
sees[(dobj)cat[(det)a],(prep)with]
sees[(nsubj)John,(dobj)cat[(amod)black]]
sees[(nsubj)John,(dobj)cat,(prep)with]
sees[(dobj)cat,(prep)with[(pobj)telescope]]
sees[(nsubj)John,(dobj)cat[(det)a]]
sees[(dobj)cat[(amod)black],(prep)with]
sees[(prep)with[(pobj)telescope[(det)a]]]

There are two possibilities for the first word in an snr-gram:

1. We can add to the first word of an snr-gram the corresponding SR tag (the name of the corresponding incoming arrow), because it always exists, or
2. We can leave the first word in an snr-gram without the SR tag, because it does not connect this word to any other element of the given snr-gram.

We choose the second option. For example, instead of the bigram *(pobj)telescope[(det)a]*, we write the bigram *telescope[(det)a]*.

In the tables above, we used dependency trees for extraction of snr-grams, but constituency trees can be used as well. There are several possibilities related to which part of the derivation history of the corresponding constituency tree should be included into the description of each relation:

– Use the derivation history that is below the node vs. above the node vs. both parts (above and below). These strategies correspond to bottom-up parsing and top-down parsing.

– Use only the left part of the rule vs. use the whole rule,
– Use only the last derivation vs. use the whole derivation chain or several last steps (say, two, three, etc.).

We present the example for (1) the whole derivation chain, (2) below the node, and (3) using the left part of the rule. Other possibilities should be tried as well in experiments for particular tasks. We start from the left element of a constituent, go up to the least common node, and then go down to the right element. At each step we take the left part of the corresponding rule. In tables 17–19 we present the snr-grams of sizes 2, 3, and 4 extracted from the example sentence.

Table 17. Snr-grams of size 2 (derivation history)

Snr-grams based on constituencies
sees[(VP,VP,PP)with]
telescope[(NP)a]
sees[(VP,NP,NP)cat]
cat[(NP)black]
with[(PP,NP)telescope]
cat[(NP)a]
sees [(S,VP,VP)John]

Table 18. Snr-grams of size 3 (derivation history)

Snr-grams based on constituencies
with[(PP,NP)telescope[(NP)a]]
sees[(VP,NP,NP)cat,(VP,VP,PP)with]
sees[(S,VP,VP)John,(VP,NP,NP)cat]
sees[(S,VP,VP)John,(VP,VP,PP)with]
sees[(VP,NP,NP)cat[(NP)a]]
sees[(VP,VP,PP)with[(PP,NP)telescope]]
cat[(NP)a,(NP)black]
sees[(VP,NP,NP)cat[(NP)black]]

Table 19. Snr-grams of size 4 (derivation history)

Snr-grams based on constituencies
sees[(VP,NP,NP)cat[(NP)a,(NP)black]]
sees[(S,VP,VP)John,(VP,VP,PP)with[(PP,NP)telescope]]

sees[(VP,NP,NP)cat[(NP)a],(VP,VP,PP)with]
sees[(S,VP,VP)John,(VP,NP,NP)cat[(NP)black]]
sees[(S,VP,VP)John,(VP,NP,NP)cat,(VP,VP,PP)with]
sees[(VP,NP,NP)cat,(VP,VP,PP)with[(PP,NP)telescope]]
sees[(S,VP,VP)John,(VP,NP,NP)cat[(NP)a]]
sees[(VP,NP,NP)cat[(NP)black],(VP,VP,PP)with]
sees[(VP,VP,PP)with[(PP,NP)telescope[(NP)a]]]

If we use SR tags, then the usefulness of snr-grams is explained by the fact that they allow to distinguish the syntactic role of each element in the n-gram, for example, "*sees[(nsubj)John]*" vs. "*sees[(dobj)cat]*", when the only difference in the verb-noun combination is the relation name. Obviously, it depends on the task if this difference is relevant or not.

In case of derivation history fragments, their function is not so clear as in case of SR tags, for example *sees[(S,VP,VP)John]* vs. *sees[(VP,NP,NP)cat]*. We can deduce that one of the fragments includes the tree root ("S"), while the other does not. We can also see how far is the distance between these two nodes in terms of the number of the applied rules and their types. Future experiments should demonstrate how useful this information is.

5 CONCLUSIONS

In this paper, we introduced and discussed the concept of the syntactic n-grams with relation names, snr-grams, which is a special type of mixed syntactic n-grams.

We have presented examples of snr-grams of various sizes, constructed for both tags of names of syntactic relations (SR tags) and for fragments of derivation history. We consider that snr-grams can be applied in many tasks of the Natural Language Processing as features for machine learning algorithms. Future experiments should confirm in which tasks their usage is beneficial.

For having the possibility of discussion of the concept of the snr-grams, we described the formalisms of dependencies and constituents used for the representation of the syntactic information and several related algorithms. We also described several issues related to the introduced in our previous works concept of syntactic n-grams.

Our future work will include analyzing the role of syntactic n-grams in different text analysis tasks as textual entailment (Pakray et al. 2010, 2011), personality detection (Poria, Gelbukh, Agarwal, Cambria & Howard 2013), sentiment analysis (Poria, Cambria, Winterstein & Huang 20104), and emotion detection (Poria, Gelbukh, Hussain, Howard, Das & Bandyopadhyay 2013; Poria, Gelbukh, Cambria, Das & Bandyopadhyay 2012). Syntactic n-grams can be very useful for the text analysis applications where insufficient training data are available, which raises the need of semi-supervised learning (Poria, Gelbukh, Hussain, Bandyopadhyay & Howard 2013).

ACKNOWLEDGMENTS This work was done under partial support of the Mexican Government (CONACYT, SNI, COFAA-IPN, SIP-IPN 20144274) and FP7-PEOPLE-2010-IRSES: "Web Information Quality–Evaluation Initiative (WIQ-EI)" European Commission project 269180.

References

1. Bolshakov, I.A., Gelbukh, A.: Computational linguistics: Models, resources, applications. IPN–UNAM–FCE, 187 pp. (2004)
2. Calvo, H., Gelbukh, A. Improving Prepositional Phrase Attachment Disambiguation Using the Web as Corpus. Lecture Notes in Computer Science, vol. 2905, pp. 604–610 (2003)
3. Cambria, E., Fu, J., Bisio, F., Poria, S.: AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 508–514 (2015)
4. Cambria, E., Poria, S., Gelbukh, A., Kwok, K.: Sentic API: A common-sense based API for concept-level sentiment analysis. In: #Microposts2014, 4th Workshop on Making Sense of Microposts at WWW 2014, CEUR Workshop Proceedings, vol. 1141, pp. 19–24 (2014)
5. Das, D., Poria, S., Bandyopadhyay, S.: A classifier based approach to emotion lexicon construction. In Natural Language Processing and Information Systems, Springer, pp. 320–326 (2012)
6. de Marneffe, M.C., MacCartney, B. Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC 2006 (2006)
7. Gelbukh, A.: Natural language processing: Perspective of CIC-IPN. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013, IEEE, pp. 2112–2121 (2013)

8. Gelbukh, A.: Syntactic disambiguation with weighted extended subcategorization frames. In: Proceedings of PACLING-99, Pacific Association for Computational Linguistics, University of Waterloo, Canada, pp. 244–249 (1999)
9. Gelbukh, A., Calvo, H. Torres, S.: Transforming a Constituency Treebank into a Dependency Treebank. Procesamiento de Lenguaje Natural, vol. 35, pp. 145–152 (2005)
10. Gelbukh, A., Kolesnikova, O.: Multiword Expressions in NLP: General Survey and a Special Case of Verb-Noun Constructions. In: Emerging Applications of Natural Language Processing: Concepts and New Research. IGI Global, pp. 1–21 (2013)
11. Gelbukh, A., Kolesnikova, O.: Semantic Analysis of Verbal Collocations with Lexical Functions. Studies in Computational Intelligence, vol. 414, Springer, 146 pp. (2013)
12. Goldberg, Y., Orwant, J.: A Dataset of Syntactic-Ngrams over Time from a Very Large Corpus of English Books. Available: http://googleresearch. blogspot.mx/2013/05/syntactic-ngrams-over-time.html (2013)
13. Ledeneva, Y., Gelbukh, A., García-Hernández, R.A.: Terms Derived from Frequent Sequences for Extractive Text Summarization. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2008. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 4919, pp. 593–604 (2008)
14. Pado, S., Lapata, M.: Dependency-based construction of semantic space models. Computational Linguistics, vol. 33, no. 2, pp. 161–199 (2007)
15. Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: Freeling 2.1: Five years of open-source language processing tools. In: Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010), ELRA (2010)
16. Pakray, P., Pal, S., Poria, S., Bandyopadhyay, S., Gelbukh, A.: JU_CSE_TAC: Textual entailment recognition system at TAC RTE-6. In System Report, Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook (2010)
17. Pakray, P., Poria, S., Bandyopadhyay, S., Gelbukh, A.: Semantic textual entailment recognition using UNL. Polibits, vol. 43, pp. 23–27 (2011)
18. Poria, S., Agarwal, B., Gelbukh, A., Hussain, A., Howard, N.: Dependency-based semantic parsing for concept-level text analysis. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 8403, Springer, pp. 113–127 (2014)
19. Poria, S., Cambria, E., Ku, L. W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Proceedings of the

Second Workshop on Natural Language Processing for Social Media, SocialNLP 2014, pp. 28–37 (2014)
20. Poria, S., Cambria, E., Winterstein, G., Huang, G.B.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems, vol. 69, pp. 45–63 (2014)
21. Poria, S., Gelbukh, A., Agarwal, B., Cambria, E., Howard, N.: Common sense knowledge based personality recognition from text. In: Advances in Soft Computing and Its Applications, MICAI 2013, Lecture Notes in Artificial Intelligence, vol. 8266, Springer, pp. 484-496 (2013)
22. Poria, S., Gelbukh, A., Cambria, E., Das, D., Bandyopadhyay, S.: Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In: Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction, SENTIRE 2012, 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW), IEEE CS Press, pp. 709–716 (2012)
23. Poria, S., Gelbukh, A., Cambria, E., Hussain, A., Huang, G.B.: EmoSenticSpace: A novel framework for affective common-sense reasoning. Knowledge-Based Systems, vol. 69, pp. 108–123 (2014)
24. Poria, S., Gelbukh, A., Hussain, A., Bandyopadhyay, S., Howard, N.: Music genre classification: A semi-supervised approach. In Pattern Recognition, Lecture Notes in Computer Science, vol. 7914, Springer, pp. 254–263 (2013)
25. Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., Bandyopadhyay, S.: Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intelligent Systems, vol. 28, no. 2, 31–38 (2013)
26. Poria, S., Ofek, N., Gelbukh, A., Hussain, A., Rokach, L.: Dependency tree-based rules for concept-level aspect-based sentiment analysis. In: Semantic Web Evaluation Challenge, SemWebEval 2014 at ESWC 2014, Greece, Revised Selected Papers. Communications in Computer and Information Science, vol. 475, Springer, pp. 41–47 (2014)
27. Sidorov, G: Syntactic Dependency Based N-grams in Rule Based Automatic English as Second Language Grammar Correction. International Journal of Computational Linguistics and Applications, vol. 4, no. 2, pp. 169–188 (2013)
28. Sidorov, G.: Non-continuous Syntactic N-grams. Polibits, vol. 48, pp. 67–75 (2013)
29. Sidorov, G.: Non-linear construction of n-grams in computational linguistics: syntactic, filtered and generalized n-grams. (in Spanish) Mexico, 166 pp. (2014)
30. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernandez, L.: Syntactic dependency-based n-grams as classification features. Lecture Notes in Artificial intelligence, vol. 7630, pp. 1–11 (2012)
31. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernandez, L.: Syntactic dependency-based n-grams: More evidence of

usefulness in classification. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2013. . Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 7816, Springer, pp. 13–24 (2013)
32. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernandez, L. Syntactic N-grams as machine learning features for natural language processing. Expert Systems with Applications, vol. 41, no. 3, pp. 853–860 (2014)

GRIGORI SIDOROV
NATURAL LANGUAGE AND TEXT PROCESSING LABORATORY,
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN,
INSTITUTO POLITÉCNICO NACIONAL (IPN),
MEXICO CITY, MEXICO
WEB: <WWW.CIC.IPN.MX/~SIDOROV>

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 49–66
Received 2/11/2014, accepted 10/12/2014, final 15/12/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Exploring Linguistic Features for Named Entity Disambiguation

SHUANGSHUANG ZHOU, CANASAI KRUENGKRAI,
NAOAKI OKAZAKI, AND KENTARO INUI

Graduate School of Information Sciences, Japan

ABSTRACT

Resolving named entities is important for a number of natural language processing applications. However, a named entity has multiple name variations while different entities could share the same surface. State-of-the-art systems are based on a global resolution method and mostly adopt link-based features that leverage relationships of co-occurring entities in the knowledge. We found that linguistic features can also significantly affect disambiguation. In this work, we try to explore important linguistic features from context, which could be the fundamental part of the combination of global resolution method and effective features. Therefore, we study and compare the effects of linguistic features in a comprehensive way. Moreover, we found effective linguistic features according to the experiment results.

KEYWORDS: *Named entity disambiguation, entity linking, feature study.*

1 INTRODUCTION

In natural language processing, named entities are important components. However, due to various ways of writing, named entities have multiple surfaces in texts, e.g., Big Blue and IBM. Moreover, different entities often share the same surface. For example, “New York” as a place name has

dozens of different referents in Wikipedia.[1] Thus, resolving name mentions to their corresponding entities is necessary. Named entity disambiguation is the task of identifying whether a mention refers to a certain entity and linking mentions to their corresponding entries in a large-scale knowledge base. Therefore, NED (Named entity disambiguation) task is also known as entity linking task.

Ji et al. [1] summarized two main processes: candidate generation and candidate ranking. Because of the variety of named entities, when given a mention, NED systems firstly often generate a candidate list contains as much as possible candidate entities. Then selecting a correct entity from the ranked candidate list by using a ranking model is the final purpose. Since the selected entity should be coherent with the context in the source document, disambiguating entities by leveraging the context information is a fundamental way [2]. Erbs et al. [3] mentioned that features for candidate ranking could be grouped into: linguistic-based (text in source document and text extracted from the KB titles) and link-based (how entities in the same context link in the knowledge).

State-of-the-art systems [4, 2] simultaneously resolve multiple entities (global inference) and mostly adopt link-based features. Those link-based features measure the relatedness of co-occurring entities in context. They assume that a candidate entity could be linked if it and its neighbor entities strongly connect in the knowledge base. For example, in the text of Figure 1, entities in context like *San Antonio Spurs* and *National Basketball Association* strongly support the [Alvin Robertson] candidate for the mention *Robertson* because they are linked in the KB. In Wikipedia, articles titled *San Antonio Spurs* and *National Basketball Association* have in-links from the article titled *Alvin Robertson*. At the same time, the article titled *Alvin Robertson* also have out-links to articles titled *San Antonio Spurs* and *National Basketball Association*.

However, when there is seldom co-occurring entities in context, linguistic information could affect disambiguation significantly [5, 6]. For example, in the text of Figure 2, words like *sophomore*, *British universities*, and *U.S. schools* suggest [University of St.Andrew] as the correct entity for *St.Andrew*. Therefore, we could capture linguistic features, such as comparing the topic distribution of source documents and the KB texts.

We find that linguistic features can locally measure the coherence between mentions and entities in context. Therefore, we study the effects of multiple linguistic features in a comprehensive way in this paper. Espe-

[1] http://en.wikipedia.org/

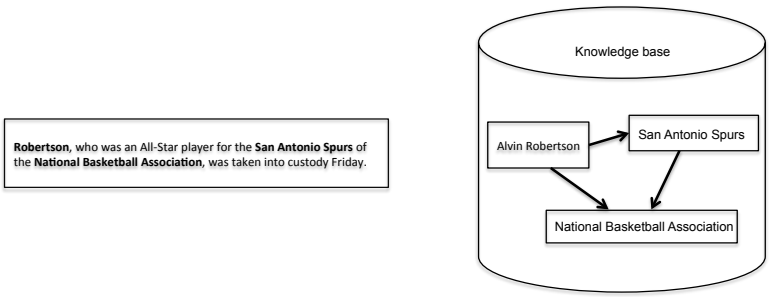


Fig. 1. Example of documents containing mentions for link-based features

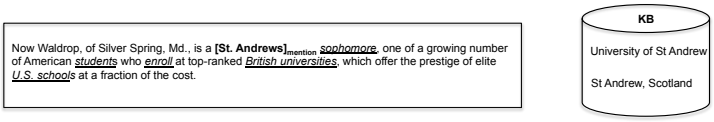


Fig. 2. Example of documents containing mentions for linguistic features

cially, we compare the effectiveness of each linguistic feature, e.g., document similarity, document topics, entity-level features, and POS features.

Moreover, several linguistic features are used as local methods by state-of-the-art global inference systems [4, 2, 7–10]. Considering their insufficient local methods, in this paper, we examine the contributions of linguistic features in order to explore more effective local methods for global inference methods.

2 Related Work

Linguistic features showed promising results in previous studies [5, 6, 11, 12], such as document similarity, word overlapping, entity-level word overlapping, document topics. However, only partial linguistic features are explored by previous work. Dredze et al. [6] captured features based on mentions, source documents and KB entries, but features about document topics are not involved. Zhang et al. [11] made big efforts on candidate selection and acronym expansion, but their disambiguation method only depended on document topics. Therefore, we summarize and refine effective linguistic features of previous work, and propose a broad range of linguistic features in this paper.

Although some previous work reviewed various ranking methods (unsupervised or supervised) and evaluation results [13, 1], they lack comparing effects of linguistic features systemically. Moreover, Garcia et al. [14] systemically reviewed and evaluated several state-of-the-art link-based approaches, but they did not mention linguistic-based context features. To our knowledge, previous work did not examine linguistic features comparatively. Therefore, we try to explore context information on the linguistic level in a comprehensive way.

On the other hand, link-based features strongly depend on the link structure of knowledge base (Wikipedia), e.g., link statistics (incoming links and outgoing links), and category information, etc. Link-based features are mostly used by global inference systems for candidate ranking. *Relatedness* is widely used by [8, 15–17, 2, 4], which is to compute the similarity between two KB entries based on the in/out links. *Relatedness* is effective to measure the relationship between candidates and co-occurring entities in context.

Linguistic features are used to measure the coherence between mentions and candidates, which are also called local methods by previous studies [2, 7, 18, 9, 10]. Combining local methods with global features or global ranking methods, the NED system performance is improved significantly [2]. Among of them, TF/IDF cosine similarity is mostly used by global inference systems for multiple purposes: ranking candidates [2, 7], filtering out noisy candidate [18], and assigning an initial confidence score for subsequent ranking phrase [9, 10]. However, TF/IDF cosine similarity is insufficient to capture the coherence between mentions and entities. Moreover, entity popularity is a salient measure of mentions and entities [8, 19–21], and it could check how likely a surface refer to an entity. Entity popularity is a strong baseline for entity linking [14]. However, this feature could ignore unpopular correct entities.

Therefore, our further motivation is to explore useful linguistic features for global inference. Based on our local features, graph-based methods are applied on short and high-coverage candidate lists, at the same time, unpopular entities can not be missed the candidate list.

3 System Architecture

The TAC KBP entity linking task provides high-quality data set and comprehensive evaluation. The data set contains a query file, a collection of source documents, and a reference KB.

In mention query files, information about one mention is given: the name surface, the document ID, and the position of this mention in the source document (UTF-8 character offsets). For example,

```
<query id="EDL14_ENG_TRAINING_3091">
    <name>St. Andrews</name>
    <docid>WPB_ENG_20101221.0031</docid>
    <beg>1123</beg>
    <end>1133</end>
</query>
```

The example texts in Figure 2 are from source documents. The TAC KBP official reference KB is extracted from an October 2008 dumps of English Wikipedia and consists of 818,741 entries.

Systems are required to generate a link-ID file, which contains pairs of a query and the resolved result (corresponding KB entry ID or NIL). For example, system should output a KB ID e.g., "E0127848" or NIL for the query "EDL14_ENG_TRAINING_3091". In this task, NIL means mentions that do not have entries in the KB. TAC KBP added the mention detection task in 2014. A system should detect possible mentions in raw documents.

We built a pipeline system for this task.[2] The system consists of basic components: mention detection, candidate generation, candidate ranking, NIL classification and NIL clustering. These five components are commonly required for performing Entity Discovery and Linking (EDL) [22]. We add the candidate pruning process after candidate generation to eliminate noisy candidates. In this work, since we want to eliminate the effect of the performance of mention detection, we train and test on the gold mention data set and start from the candidate generation phase. In order to simplify the evaluation, we remove the NIL clustering process.

3.1 *Candidate Generation*

In the candidate generation phase, we need a high-recall candidate list for each mention. In this phase, recall means the percentage of mentions that have the correct entity in the candidate list. If the correct entity does not exist in the candidate list, the ranking process will be in vain. We first group mentions in the source document to handle misspelling, abbreviation, and partial names. For example, the candidate mentions *Gretzy* and

[2] We submitted the system to TAC KBP 2014 entity discovery and linking shared task.

Wayne Gretzky occur in the same source document, and they likely refer to the same entity. If we search candidates by using both of them, the possibility of correct entity appearing in the candidate list of *Gretzy* could be increased.

Moreover, we construct a name variation database, SurfaceSet, by extracting entity title-surface pairs from various Wikipedia sources, such as disambiguation pages, redirection pages, and anchor texts. For example, we extracted name variations like *Barcodes*, *Toon*, *mags*, *magpies*, and *Newcastle* for *Newcastle United F.C.*, a famous England football club. SurfaceSet contains 548,084 entities and 2,080,491 surfaces.

We process one mention at one time. For each mention, we search both the original mention and the same group mentions. We achieved 98.43% recall on the training set. The average number of candidate per list is 245.

3.2 *Candidate Pruning*

Note that the initial candidate lists are too noisy because we want to find as many as possible candidates in the previous phase. Ranking document similarities between source documents and wiki texts is a simple and efficient way to eliminate noisy candidates. According to our preliminary experiment, we found that Latent Semantic Index (LSI) is superior to TF/IDF cosine similarity. LSI achieved 97.39% recall on the training set while TF/IDF got 74.84%. We apply Latent Semantic Index (LSI) to rank each candidate list and retain the top 50 candidates as the final candidate list. We use an off-the-shelf tool, gensim [23]. The average number of candidates per list is 41.

3.3 *Candidate Ranking*

In the candidate ranking phase, we formulate the ranking problem similar to [5, 24]. We generate a score function $f(m, e_i)$ based on features that extracted from the mention m and a candidate e_i. We select a candidate entity from candidate list E for a mention according to the highest score:

$$e = \text{argmax}_{e_i} f(m, e_i), e_i \in E \tag{1}$$

Therefore, the correct entry e for a mention m obtain a higher score than all other candidates $\hat{e} \in E, \hat{e} \neq e$. We use SVM$^{\text{rank}}$ [25] with the linear kernel to handle the optimization problem.

3.4 *NIL Classification*

We use heuristic rules to determine the final label for a mention. Mentions are labeled as NIL if there is no candidate in the candidate list or the ranking score of the top 1 candidate is below a threshold.

4 FEATURE STUDY

We extract multiple features for candidate ranking. First, we extract basic features from mention surfaces. In order to explore linguistic information in context, we categorize those linguistic-based features into several groups.

4.1 *Basic Features*

We focus on the surface properties of the KB title and the mention surface. The IsAcronym and IsAbbrMatch features [6] capture characteristics of acronyms. For example, given a mention *WTO*, acronym features can detect **W***orld* **T***rade* **O***rganization*. The SurfaceSimScore and EqualWordNumSurface features [26] calculate how similar is the mention surface to the KB title. The TokenLenInCandidate and CharLenInCandidate [12] count the terms and characters of the KB titles. We also incorporate other similarity features used in previous work [12, 27], such as dice coefficient scores and jaccard index scores. We summarize the basic features in Table 1.

4.2 *Linguistic Context Features*

We extract linguistic information from both mention source documents and texts of knowledge base entries (candidates) for disambiguation.

Title appearance Title appearance features [12] are related with the appearance of a candidate title in the source document, or the appearance of mentions in candidate texts. For example, if a given mention is the family name of a person, e.g., *Daughtry*, the title of a candidate, e.g., *Chris Daughtry*, may appear in the source document. Similarly, this given mention *Daughtry* may occur in the text of KB entry *Chris Daughtry*. Among them, a salient feature [12] detects disambiguators in candidate titles, e.g., *magazine* in People (magazine) and *basketball* in Maurice Williams (basketball).

Table 1. Basic Features of Candidate Ranking Module

Feature	Description
SurfaceSimScore	Levenshtein edit distance between the KB title and the mention surface
EqualWordNumSurface	Maximum of count of exact matches between mentions in the same group and the KB title
HasQueryGroup	Whether the KB title belongs to a mention group
QueryGroupMatch	Whether the KB title matches any surface in the same group
QueryGroupOverlap	Whether a surface in the same group is substring of the KB title, or vice versa
QueryGroupMaxSim	Maximum similarity between the KB title and surfaces in the same group
TokenLenInCandidate	Term count in the KB title
CharLenInCandidate	Characters count in the KB title
IsAcronym	Whether the mention surface is an acronym
IsAbbrMatch	Whether the capital character of the KB title match any surface in the same group
DiceTokenScore	Maximum value of the dice coefficient between the KB title token set and the surface token set
DiceToken	Whether DiceTokenScore is above 0.9
JaccardTokenScore	Maximum value of jaccard index between the KB title token set and the surface token set
DiceCharacterScore	Maximum value of dice coefficient between the KB title character set and the surface character set
DiceCharacter	Whether DiceCharacterScore is above 0.9
DiceAlignedTokenSocre	Maximum character dice coefficient of left and right aligned token sets
DiceAlignedToken	Whether DiceAlignedTokenScore is above 0.9
DiceAligned-CharacterSocre	Maximum character dice coefficient of left and right aligned character sets
DiceAlignedCharacter	Whether DiceAlignedCharacterSocre is above 0.9

Document similarity We use two measures to compare the text similarity between source documents and KB texts: cosine similarity with TF/IDF [26] and dice coefficient [27] on tokens. Since the first paragraphs of KB and text surrounding mention are supposed to be more informative, we consider using different ranges of source documents and KB texts. We divide text in a source document into local text (window size = 50 tokens)

and global text (the whole source document), and use the first paragraph and the whole KB text receptively.

Entity mention occurrence Named entities in mention context are more salient than common words. This feature is used in [6], which could capture the count of co-occurring named entities between source documents and KB texts. For example, for a given mention *Obama*, the named entities *White House* and *United States* may appear in both the source document and the KB text if it refers to the American president *Barack Obama*.

Entity fact The infobox of KB contains important attributes of entries. For example, for entity *Apple Inc.*, we can extract attributes, such as Founder (*Steve Jobs*) and CEO (*Tim Cook*). Therefore we extract fact texts from KB and check whether fact texts are in source documents, which is inspired by [6]

Document topics Semantic information cannot be detected by simply counting occurrences of tokens, n-grams, and entities. Therefore we use topic models to discover the implicit topics of source documents and KB texts. We train LDA (Latent Dirichlet Allocation) model with gensim [23], which provides a fast online LDA model. We treat each KB entry as one document and use two different corpus for training. Zhang et al. [11] trained a topic model on the KBP knowledge base, we additionally train another topic model on the latest wikidump.[3] The KBP knowledge base is a partial KB and contains about one third of Wikipedia entities. We use two similarity measures to check the topic similarity between source documents and KB entries including cosine similarity and Hellinger distance. We also generate topics of partial text surrounding mention as the local topics to compare with using the whole source document (global topics).

Similarity of part-of-speech tokens We hypothesize that nouns and verbs compared to other type of words could contribute more on disambiguating. Therefore we collect this two type of tokens in context and calculate cosine similarity with TF/IDF weighting respectively.

Entity type We use entity type matching to detect whether the KB entity type is identical to the mention entity type, which is similar with [6]. For example, the mention *St.Andrew* is an ORG (Organization) entity in the first text in Figure 2. The candidate *University of St.Andrew* (ORG) is

[3] http://dumps.wikimedia.org/enwiki/20140707/enwiki20140707-pages-articles.xml.bz2

more likely than *St.Andrew, Scotland* (GPE) because of entity type matching. Therefore, we should predict named entity types for both non-NIL mentions and NIL mentions in the final output results. Since the KBP KB provides entity type information, we concern that it is more credible to predict non-NIL mention types by using KBP KB labeled type. However there are almost 64.9% unknown entities in the official KBP KB. It means that we need to re-tag remaining unknown entities. Unlike [6], we use the re-tag entity types according to our re-tagging results.

Clarke et al. [28] classified unknown type entities based on the infobox class. They also found that matching between infobox classes and entity types approximately has no ambiguity. Unlike [28] classified infobox class using learning method, we resolved around 2370 infobox classes manually. Our re-tagged result contains four types: PER, ORG, GPE, and MISC. Table 2 shows the comparison before and after re-tagging process.

Table 2. Entity types before and after re-tagging

Type	KBP KB	Our System
PER	14.0%	23.5%
ORG	6.8%	12.3%
GPE	14.2%	22.0%
UKN	64.9%	0.0%
MISC	0.0%	42.2%

5 EVALUATION

5.1 *Data Set and Evaluation Metric*

We use the training data from the 2014 TAC KBP Entity Discovery and Linking (EDL) track [22]. The TAC data set consists of 5878 mentions over 158 documents. The statistics of the data set is shown in Table 3. We use the gold mention query file of the data set.

Our evaluation metric is micro-averaged accuracy, which is used in TAC KBP 2009 and 2010 entity linking task [13]. The metric is computed by

$$Accuracy_{micro} = \frac{NumCorrrect}{NumMentions} \tag{2}$$

Table 3. Statistics of 2014 TAC KBP Data set

	PER	ORG	GPE	Total
NIL	1819	591	216	2626
Non-NIL	1390	709	1153	3252
Total	3209	1300	1369	5878

5.2 *Experiment*

Since we focus on the ranking performance of each group of linguistic-based context features, we compute the accuracy of mentions system resolved. In order to eliminate the effect of feature combination, we add only one feature group to the basic feature group each time. We performed 5-fold cross-validation on the training set. Table 4 shows micro-averaged accuracies of feature addition experiments.

Table 4. Feature additive test results

Feature Group	**Non**	**NIL**	**ALL**
Basic	0.5910	0.7000	0.6394
Title Appearance	0.6138	0.7086	0.6558
Entity Fact	0.6024	0.6664	0.6306
Entity Mention Occurrence	0.6134	0.7668	0.6814
Document Similarity	0.6594	0.7733	0.7059
Document Similarity (LOCAL)	0.6422	0.7403	0.6860
Document Similarity (GLOBAL)	0.6474	0.7881	0.7096
Document Topic	0.6322	0.6912	0.6580
Document Topic (WIKI)	0.6224	0.6912	0.6528
Document Topic (KBP)	0.6280	0.6880	0.6544
Similarity of POS	0.6224	0.7420	0.6754
Similarity of POS (Noun)	0.6236	0.7364	0.6736
Similarity of POS (Verb)	0.5986	0.6970	0.6416
Type	0.5908	0.7030	0.6400
All Features	0.7330	0.7454	0.7378

In Figure 3, we consider subsets of mentions by entity type. We compare the entity linking performance on three types respectively.

In order to clarify feature effects, we divide features into more fine-grained groups, such as local topics (DT_WIKI_LOC, DT_KBP_LOC), global topics (DT_WIKI_GLO, DT_KBP_GLO), and document similarity by using the first paragraph of KB texts (DS_CON_FIR) or the whole KB

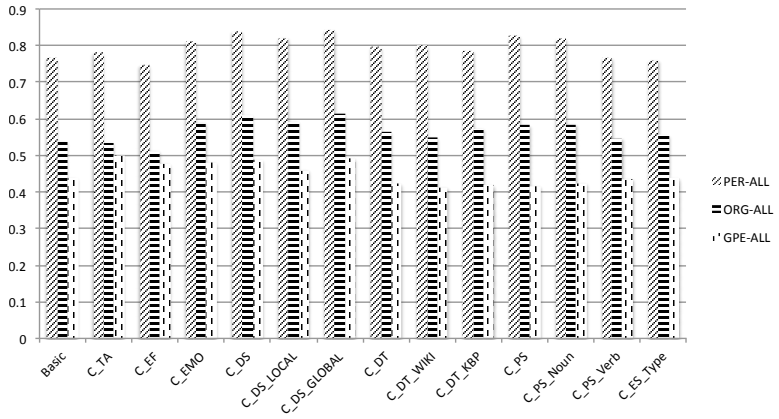


Fig. 3. Entity linking performance on PER, ORG, GPE entities

texts (DS_CON_ALL). Table 5 shows the increment of each fine-grained feature group to basic features on non-NIL mentions before NIL classification processing, and feature group names are capitalized referring to Table 4.

Moreover, we plan to check whether our current linguistic features can be used as local methods for global inference methods.Therefore, we compare the performance of our features with two local methods of previous work: 'TF/IDF cosine similarity' and 'Entity Popularity'. We evaluate the accuracy on the TAC KBP 2014 test data set. Since considering using local methods to filter out noisy candidates for global inference methods, we calculate the percentage of correct entities on different positions. The results are shown in Table 6. Acc@1, Acc@5, and Acc@10 mean correct entities on top1 position, within the top5 results, and within the top10 results.

We only focus on the performance of the in-KB entities because correct entities are provided in the data set. Similar with [20], we retrieve a mention with the Freebase search API,[4] and we select an entity which has the highest popular score of all the returned entities. We found that our features overcome 'TF/IDF cosine similarity' and 'Entity Popularity' in all the three cases.

[4] https://developers.google.com/freebase/v1

Table 5. Accuracy increment on non-NIL mentions before NIL classification

Fine-grained Feature Group	Accuracy Increment
C_DS_LOCAL	0.0736
C_DS_GLOBAL	0.1044
C_DS_CON_FIR	0.0214
C_DS_CON_ALL	0.0726
C_DT	0.0582
C_DT_WIKI	0.0338
C_DT_KBP	0.0576
C_DT_WIKI_GLO	0.0576
C_DT_WIKI_LOC	0.0344
C_DT_KBP_GLO	0.0534
C_DT_KBP_LOC	0.0510
C_PS_Noun	0.0658
C_PS_Verb	0.0150

Table 6. Accuracy at different positions

	Acc@1	Acc@5	Acc@10
TF/IDF cosine similarity	0.5066	0.8204	0.8910
Entity Popularity	0.3710	0.5453	0.6124
All Features	0.8264	0.9418	0.9492

6 Discussion and Future Work

6.1 *Overall Feature Effects*

Basic features only include features related to surface similarity, which is not effective enough to find correct entities. Features based on document similarity (both words and part-of-speech levels), named entities co-occurrence, and document topics contribute the most gains.

Document similarity In both document similarity and document topics, global features are better than local features. Since we leverage measures based on bag-of-words calculation, the larger text of context contains more co-occurring words than the window-size context. Although we suggest that the first paragraph in the KB is much informative, using the whole KB text (DS_CON_ALL) is much better than only using the first paragraph (DS_CON_FIR). We found that, in the KBP KB, several first paragraphs of KB texts are very short, sometimes only one sentence. For example, for *Jeff Perry (American actor)*, there is only one sentence,

"Jeff Perry (born August 16, 1955 in Highland Park, Illinois) is an American character actor."

We found that around 28.74% entries of the KBP KB contain one simple sentence in the first paragraph.

Document topics Moreover, based on the results in Table 5, the increment of global topics is more than that of local topics by 0.024 (KBP corpus). Since the distribution of partial document topics is inconsistent with document topics, global topics can better represent the semantic context of a mention.

Although the KBP KB contains around one third entities of Wikipedia, the performance on the KBP KB corpus is better because we use the KBP KB as the entities database. We found that words of KBP KB topics could represent source document better than using the Wikipedia corpus for some entities. For example, *Salvador Dali* entity is a painter, who is also known for writing and film. Words of top topics are given by the KBP LDA corpus of this entity are *film*, *book*, *album*, *play*. However, words given by the Wikipedia LDA corpus are *Louisiana*, *disease*, *species*, and so on. The Wikipedia LDA corpus is not well-built, which may also affect the performance, because we follow an off-the-shelf training process.[5]

However, from Table 4 one can see that the performance on Wikipedia corpus is slightly effective on NIL by 0.003.

Similarity of POS tokens In Table 4, we found that nouns are more informative than verbs by around 0.3. Nouns contain more information than verbs because named entities are more salient.

6.2 *Feature Effects on Different Entity Types*

Figure 3 shows the performance on PER entities is much better than ORG and GPE entities by more than 20 percentage. It reveals that our features are biased toward PER entities and ORG and GPE mentions are difficult to resolve. After the error analysis, we found that simple string matching linguistic features fail to disambiguate ORG and GPE entities because of multiple name variation, especially the confusion between different entity types. For example, city names could be part of sport teams (*Orlando* is short for *Orlando Magic*) and people names could be part of company names (*Disney* is short for *Walt Disney Company* or *Walt Disney Animation Studio*). Moreover, the amount of PER mentions is two times larger than the amount of ORG mentions or GPE mentions in our data set.

[5] https://radimrehurek.com/gensim/wiki.html

6.3 *Future Work*

We compared the performance of current features with systems from the 2014 EDL Diagnostic task [22]. The accuracy on all mentions of our current system could beat the median system by 0.5, but we still have a huge gap with the best system. Even for some top systems from the 2014 EDL workshop [22] , performance on ORG and GPE entities are still much worse than PER entities. It should be an important future work to discover effective features which can solve ORG and GPE entities better.

Since we use a simple heuristic method to classify non-NIL and NIL mentions, the accuracy significantly drops after NIL classification process. In future work, we will explore effect features on determining NIL entities and improve the NIL classification method. Moreover, we plan to combine linguistic features with link-based methods to further improve our system.

REFERENCES

1. Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J.: Overview of the tac 2010 knowledge base population track. In: Third Text Analysis Conference (TAC 2010). (2010)
2. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. (2011) 1375–1384
3. Erbs, N., Zesch, T., Gurevych, I.: Link discovery: A comprehensive analysis. In: 2011 Fifth IEEE International Conference on Semantic Computing (ICSC), IEEE (2011) 83–86
4. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2011) 782–792
5. Bunescu, R.C., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of EACL. Volume 6. (2006) 9–16
6. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population. In: Proceedings of the 23rd International Conference on Computational Linguistics. (2010) 277–285
7. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: EMNLP-CoNLL. Volume 7. (2007) 708–716
8. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM (2008) 509–518

9. Guo, Z., Barbosa, D.: Robust entity linking via random walks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM (2014) 499–508
10. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: A graph-based method. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM (2011) 765–774
11. Zhang, W., Sim, Y.C., Su, J., Tan, C.L.: Entity linking with effective acronym expansion, instance selection, and topic modeling. In: Proceedings of IJCAI. Volume 2011. (2011) 1909–1914
12. Graus, D., Kenter, T., Bron, M., Meij, E., De Rijke, M.: Context-based entity linking – University of Amsterdam at TAC 2012. In: Proceedings of the Fifth Text Analysis Conference (TAC 2012): November 5–6, 2012, National Institute of Standards and Technology, Gaithersburg, Maryland, USA. (2012)
13. McNamee, P., Dang, H.T.: Overview of the tac 2009 knowledge base population track. In: Text Analysis Conference (TAC). Volume 17. (2009) 111–113
14. Garcıa, N.F., Fisteus, J.A., Fernández, L.S.: Comparative evaluation of link-based approaches for candidate ranking in link-to-Wikipedia systems. Journal of Artificial Intelligence Research **49** (2014) 733–773
15. Han, X., Zhao, J.: Named entity disambiguation by leveraging Wikipedia semantic knowledge. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM (2009) 215–224
16. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of Wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (2009) 457–466
17. Guo, Y., Tang, G., Che, W., Liu, T., Li, S.: Hit approaches to entity linking at tac 2011. In: Proceedings of the Fourth Text Analysis Conference (TAC 2011). (2011)
18. Guo, S., Chang, M.W., Kiciman, E.: To link or not to link? a study on end-to-end tweet entity linking. In: HLT-NAACL. (2013) 1020–1030
19. Ferragina, P., Scaiella, U.: Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM (2010) 1625–1628
20. Alhelbawy, A., Gaizauskas, R.: Graph ranking for collective named entity disambiguation. In: Proceedings of the 52nd Annual Meeting of the ACL. (2014) 75–80
21. Alhelbawy, A., Gaizauskas, R.: Collective named entity disambiguation using graph ranking and clique partitioning approaches. In: Proceedings of COLING. (2014) 1544–1555
22. Ji, H., Dang, H., Nothman, J., Hachey, B.: Overview of tac-kbp2014 entity discovery and linking tasks. In: Proceedings of Text Analysis Conference. (2014)

23. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC: Workshop on New Challenges for NLP Frameworks. (2010) 45–50
24. McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., Mayfield, J., Piatko, C., Rao, D., Yarowsky, D., Dreyer, M.: Hltcoe approaches to knowledge base population at tac 2009. In: Proceedings of TAC 2009. (2009)
25. Joachims, T.: Training linear SVMs in linear time. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (2006) 217–226
26. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to link entities with knowledge base. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 483–491
27. Dietz, L., Dalton, J.: Acrossdocument neighborhood expansion: Umass at tac kbp 2012 entity linking. In: Proceedings of TAC 2012. (2012)
28. Clarke, J., Suleiman, Y.M.G., Murgatroyd, S.Z.D.: Basis technology at tac 2012 entity linking. In: Proceedings of TAC 2012. (2012)

SHUANGSHUANG ZHOU
INUI-OKAZAKI LAB.,
GRADUATE SCHOOL OF INFORMATION SCIENCES,
6-6 ARAMAKI AZA AOBA, AOBAKU, SENDAI, MIYAGI 980-8579,
JAPAN
E-MAIL: <SHUANG@ECEI.TOHOKU.AC.JP>

CANASAI KRUENGKRAI
INUI-OKAZAKI LAB.,
GRADUATE SCHOOL OF INFORMATION SCIENCES,
6-6 ARAMAKI AZA AOBA, AOBAKU, SENDAI, MIYAGI 980-8579,
JAPAN
E-MAIL: <CANASAI@ECEI.TOHOKU.AC.JP>

NAOAKI OKAZAKI
INUI-OKAZAKI LAB.,
GRADUATE SCHOOL OF INFORMATION SCIENCES,
6-6 ARAMAKI AZA AOBA, AOBAKU, SENDAI, MIYAGI 980-8579,
JAPAN
E-MAIL: <OKAZAI@ECEI.TOHOKU.AC.JP>

KENTARO INUI
INUI-OKAZAKI LAB.,
GRADUATE SCHOOL OF INFORMATION SCIENCES,
6-6 ARAMAKI AZA AOBA, AOBAKU, SENDAI, MIYAGI 980-8579,
JAPAN
E-MAIL: <INUI@ECEI.TOHOKU.AC.JP>

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 67–94
Received 14/05/2014, accepted 12/08/2014, final 27/10/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Discriminative Ability of WordNet Senses on the Task of Detecting Lexical Functions in Spanish Verb-Noun Collocations

OLGA KOLESNIKOVA

Instituto Politécnico Nacional, Mexico

ABSTRACT

Collocations, or restricted lexical co-occurrence, are a difficult issue in natural language processing because their semantics cannot be derived from the semantics of their constituents. Therefore, such verb-noun combinations as "take a break," "catch a bus," "have lunch" can be interpreted incorrectly by automatic semantic analysis. Since collocations are combinations frequently used in texts, errors in their analysis cannot be ignored. The quality of analysis of collocations can be improved if they are annotated with lexical functions that represent semantic classes of collocations. In this work, we study how WordNet senses viewed as sets of hypernyms can distinguish lexical functions of Spanish verb-noun collocations in experiments with supervised machine learning methods. We show that WordNet senses discriminate lexical functions to different degrees depending on the function, and this phenomenon can be used to evaluate the quality of word sense definitions as well as to measure similarity of various senses of a word and the correlation between word senses and lexical functions.

1 INTRODUCTION

Collocation is a word combination whose semantics cannot be derived from typical meaning of each component word. Very often collocations

are combinations of two words. For example, *have* is commonly interpreted as ‘possess’, and the meaning of *lunch* is ‘midday meal’. However, *have lunch* cannot be understood as ‘possess a midday meal’ since the noun *lunch* preserves its typical meaning of a midday meal, but the verb *have* acquires another meaning, ‘consume’. Therefore, *have lunch* is correctly interpreted as ‘consume a midday meal’. Due to this peculiarity of the verb-noun combination *have lunch*, it is termed collocation to distinguish it from other syntactically similar phrases termed free word combinations whose meaning can be represented as a sum of meanings of their component words: *have a daughter*, *have a book*, *have a nice house*, etc.

Recognition and correct interpretation of collocations is a big challenge in natural language processing (NLP). Errors in semantic analysis of collocations cannot be easily ignored due to their high frequency: about 43% of entries in the English WordNet are collocations [c, d]; also, depending on a specific domain, collocations can comprise up to 85% of vocabulary in texts [11]. Therefore, adequate detection and adequate processing of collocations plays a very significant role in all natural language processing applications that include a module for performing semantic analysis of texts to various degrees of granularity.

As previously mentioned, in the verb-noun collocation *have lunch*, the noun *lunch* preserves its typical sense, but the verb *have* changes its meaning. Why is it so? It seems from the set of synonyms of *have* which is {*command*, *enjoy*, *hold*, *own*, *retain*} (taken from Merriam-Webster Thesaurus online, http://www.merriam-webster.com), *lunch* chooses a combination with *have* in order to generate the meaning ‘consume food in the afternoon’. Notice that the noun *food* prefers another verb, *take*, to express the same semantics of ‘consuming a solid substance used for nourishment’. So, two different verbs *have* and *take* express the same meaning but each of them in combination with different nouns. Such usage is also termed ‘restricted lexical co-occurrence’ meaning that we can say *have lunch* but not **take lunch* in the sense of eating it. Therefore, in a collocation, one word “chooses” another one; in the case of verb-noun collocations, a noun chooses a verb and modifies its meaning. The noun is called the base of a collocation, and the verb is called the collocate. In this paper we study only verb-noun collocations.

There are many state-of-the-art methods for automatic detection and extraction of collocations. Such techniques produce lists of collocations. Lists of collocations would be more useful if collocations were tagged

with semantic information. In other words, semantic analysis of collocations is needed in order to interpret their meaning correctly.

In some cases such collocations correspond to so-called concepts [14]. In sentiment analysis and opinion mining it is very important to identify such concepts [15, 21]. In standard sentiment lexicons, collocations are either ignored or assigned neutral polarity. However, modern research shows that collocations carry meaning and sentiment. They play major role in, for example, contextual polarity shifting [3]. Collocations are also useful to understand emotions [4]. Some researchers use concept vectors instead of bag-of-words models [2, 19].

In this paper, we consider semantic analysis of a certain type, namely, semantic classification of collocation according to lexical functions. This task can also be viewed as automatic detection of lexical functions in collocations. The concept of lexical function is a formalism within the Meaning-Text Theory [7, 8] and is explained in the section that follows (Section 2, Lexical Functions).

In this paper, we study how and with what precision lexical functions can be distinguished by WordNet senses [9]. WordNet senses can be characterized by many features included in this ontology: glosses (definitions of words), synsets (words synonymous to a given word), relations (hypernymy, hyponymy, antonymy, meronymy, troponymy, entailment), sentence frames, examples of word usage, etc. We represent each word sense by a set of all its hypernyms. A hypernym is a word whose meaning is more generic than the meaning of a given word; for example, *furniture* is a hypernym of *chair*.

We chose the hypernymy relation to represent word senses because this feature has been used in the state of the art research on automatic detection of lexical functions, so there is enough experimental data published in literature to compare our results with. Also, as it will be seen in the next section, lexical function is a tool designed to generalize semantics of collocations, so supposedly hypernyms as words with more general semantics can be helpful in lexical function identification.

In this research we analyze the ability of hypernyms of verb-noun collocation constituents to discriminate lexical functions. What hypernyms and corresponding to what word senses discern lexical functions with a higher precision? This is the basic issue we deal with here. Also, we consider another issue: how variations in lexical function detection precision depending on WordNet senses can be understood and

interpreted with respect to such characteristic of collocations as relative or non-compositionality [26].

For automatic detection of lexical functions, we use a dataset of Spanish verb-noun collocations [5, 23] and supervised machine learning techniques.

The rest of the paper is organized as follows. Section 2 discusses the concept of lexical functions as semantic classes of collocations. Section 3 shows some applications of lexical functions in natural language processing. In Section 4 we review state of the art research on automatic detection of lexical functions. In Section 5 we define the problem and questions we deal with in this research. Section 6 describes our experiments, their results are discussed in Section 7, and Section 8 presents conclusions and outlines future work.

2 Lexical Functions

Lexical function (LF) is a formal concept proposed within the Meaning-Text Theory [7, 8] to generalize and represent both semantic and syntactic structure of a collocation. LF is similar to a mathematical function and has the form

$$\mathrm{LF}(w_0) = \{w_1, w_2, \dots, w_n\}, \quad (1)$$

where w_0 is the LF argument which is the base of a collocation, and the LF value is the set $\{w_1, w_2, \dots, w_n\}$ whose elements are words or word combinations $w_i, 0 < i \leq n$, which is/are collocate/s of a given base. In the present research we consider only verb-noun collocations and, respectively, verb-noun lexical functions, so applying the above formula to this particular group of collocations we have w_0 to denote a noun (base of a collocation) and the set $\{w_1, w_2, \dots, w_n\}$ will now include only one element w_1 which is a verb (collocate in a verb-noun collocation). Thus we will study lexical functions of the following type:

$$\mathrm{LF}\colon \mathrm{N} \rightarrow V, \quad (2)$$

where *N* is a set of nouns in which each noun functions as a base in a verb-noun collocation, and *V* is a set of all verb collocates.

LF in the Formulas 1 and 2 represents the generalized semantics of groups of verbal collocates on the one hand, and on the other hand, captures the basic syntactic and predicate-argument structure of

sentences in which a collocation belonging to such group is used. Therefore, a lexical function can be viewed as a formal representation of semantic, syntactic, and governing patterns in which verb-noun collocations participate. We will explain and illustrate this formalism with some of most common lexical functions:

- Oper1, from Latin *operari = do, carry out*, formalizes the action of carrying out of what is denoted by the noun (LF argument). Integers in the LF notation are used to specify the predicate-argument and syntactic structure. In Oper1, 1 means that the word used to lexicalize the semantic role of agent of the action denoted by the verb (agent is considered the first argument of a verb) functions as the grammatical subject in a sentence, so Oper1 represents the pattern *Agent performs* w_0 (w_0 is the argument of a lexical function, see Formula 1). For example, Oper1(*decision*) = *make*, and in the sentence *The president made a decision*, *president* is the agent and its syntactic function is subject. Other verb-noun collocations which can be covered by Oper1 are *pursue a goal*, *make an error*, *apply a measure*, *give a smile*, *take a walk*, *have lunch*, *deliver a lecture*, *make an announcement*, *lend support*, *put up resistance*, *give an order*.
- Oper2 has the meaning 'undergo, meet' and represents the pattern *Patient undergoes* w_0, for example, *suffer a change*, *receive support*, *receive an order*, *meet resistance*.
- Func0, from Latin *functionare = to function*, represents the meaning 'happen, take place'. The noun argument w_0 of Func0 is the name of an action, activity, state, property, relation, i.e., it is such a noun whose meaning is or includes a predicate in the logical sense of the term thus presupposing arguments. Zero in Func0 means that the argument of Func0 is the agent of the verb and functions as the grammatical subject in a sentence. Therefore, Func0 represents the patterns w_0 *occurs*. For example, *snow falls*, *silence reigns*, *smell lingers*, *time flies.*
- Real1, from Latin *realis = real*, means 'to use the noun argument w_0 according to its destination', 'to do with w_0 what one is supposed to with w_0', 'to do with regard to w_0 what is normally expected of the agent', so Real1 represents the pattern *Agent acts according to* w_0: *do one's duty*, *fulfill an obligation*, *keep a secret*, *follow a principle*, *obey a command*.

Each lexical function discussed above represents one simple meaning or a single semantic unit, so such functions are called simple. There are lexical functions that formalize combinations of unitary meanings; they are called complex lexical functions. Now we will consider some of them:

- IncepOper1 is a combination of the semantic unit 'begin', from Latin *incipere*, and Oper1 presented above. This LF has the meaning 'begin doing something' and represents the pattern *Agent begins to do the <noun>*: *to open fire on …, to acquire popularity, to sink into despair, to take an attitude, to obtain a position, begin negotiations, fall into problems*.
- ContOper1 combines the meaning 'continue', from Latin *continuare*, with Oper1. It represents the pattern *Agent continues to do* w_0, for example, *maintain enthusiasm, maintain supremacy, keep one's balance*.
- Caus, from Latin *causare*, represents the meaning 'cause, do something so that w_0 begins occurring'. Caus is used only in combinations with other LFs. So CausFunc0 means 'to cause the existence of w_0' and represents the pattern *Agent does something such that* w_0 *begins to occur*: *bring about the crisis, create a difficulty, present a difficulty, call elections, establish a system, produce an effect*. CausFunc1 represents the pattern *Non-agent argument does something such that* w_0 *begins to occur*, for example, *open a perspective, raise hope, open a way, cause a damage, instill a habit into somebody*.

3 Application of Lexical Functions in Natural Language Processing

Lexical functions possess a number of important properties which make them an effective tool for natural language processing. First, LFs are universal; it means that a significantly little number of LFs (about 70) represent the fundamental semantic relations between words in the vocabulary of any natural language and the basic semantic relations which syntactically connected word forms can obtain in a text. Secondly, LFs are characteristic for idioms in many natural languages and can serve as a typology for classification of idioms, collocations, and other types

of restricted lexical co-occurrence. Thirdly, LFs can be paraphrased. For example, the LFs Oper and Func can form combinations with their arguments which are synonymous to the basic verb like in the following utterances: *The government controls prices – The government has control of prices* – The *government keeps prices under control – The prices are under the government's control.*

LFs can be used to resolve syntactic ambiguity. In such cases, syntactically identical phrases are characterized by different lexical functions which serve as a tool for disambiguation.

For example, consider two phrases: *support of the parliament* and *support of the president.* In the first phrase *support* is the object, but in the second phrase *support* functions syntactically as the subject and semantically as the agent. The surface phrase structure in both cases is identical: *support* + *of* + noun; this fact causes syntactic ambiguity and due to it both phrases may have both meanings: 'support given by the parliament (by the president)', which syntactically is the subject interpretation with the agentive syntactic relation between *support* and the subordinated noun, and 'support given to the parliament (to the president)' which syntactically is the object interpretation with the first completive syntactic relation between *support* and the subordinated noun. This type of ambiguity is often extremely difficult to resolve, even within a broad context. LF verbs can be successfully used to disambiguate such phrases because they impose strong limitations on the syntactic behavior of their arguments in texts.

Now let us view the same phrases in a broader context. The first example is *The president spoke in support of the parliament,* where the verb *to speak in* is Oper1 of the noun *support*, i.e., Oper1(*support*) = *speak in*. Oper1 represents the pattern *Agent performs* w_0 (where w_0 is the argument of Oper1), so *the president* is interpreted as the agent, and *support* as the object. Therefore, *the president spoke in support of the parliament* can only be interpreted as describing the support given to the parliament, with *parliament* having the syntactic function of the complement of *support*.

On the other hand, verbs of Oper2 participate in another pattern: *Patient undergoes* w_0. So Oper2 verb is by definition a verb whose grammatical subject represents the patient of w_0 and in the utterance *the president enjoyed* (Oper2) *the support of the parliament,* the phrase *the support of the parliament* implies the support given to the president by

the parliament, with *parliament* having the syntactic function of the agentive dependent of the noun *support*.

LFs can also be used in computer-assisted language learning. It is a well-known fact in second language teaching practice that collocations are difficult to master by learners, so learner's speech often sounds unnatural due to errors in restricted lexical co-occurrence. To deal with this issue, a lexical function dictionary can be used whose advantage is that it includes the linguistic material on word combinations which is absent in word dictionaries.

LFs can be used in machine translation due to their semantic universality and cross-linguistic idiomaticity. These characteristics make LFs an ideal tool for selecting idiomatic translations of set expressions in a machine translation system. *They took a walk after lunch* is translated into Spanish by Google Translate as *Tomaron un paseo después del almuerzo* (translated on May 6, 2015). In English, Oper1(*walk*) = *take*, but in Spanish Oper1 of the argument *paseo* (English *walk*) is *dar* (English lit. *give*). So Oper1(*paseo*) = *dar*, however, the system translated the collocation *take a walk* literally as *tomar paseo*, since *take* is literally *tomar* in Spanish. Therefore, a module that annotates word combinations with lexical functions can be included in any machine translation system to improve the quality of translation of collocations and idiomatic expressions.

Patterns corresponding to LFs can be used in other natural language processing tasks: parsing, semantic role tagging, text analysis, etc. For example, LF patterns can be used as templates for generating grammatical sentences in automatic text generation.

4 Automatic Detection of Lexical Functions

There have been made a few attempts to detect LFs automatically. Wanner [28] approached automatic detection of LFs as a task of automatic classification of collocations according to LF typology. He applied Nearest Neighbor machine learning technique to classify Spanish verb-noun pairs according to nine LFs selected for the experiments. The distance of candidate instances to instances in the training set was evaluated using path length in hypernym hierarchy of the Spanish part of EuroWordNet [25, 27] corresponding to each verb and noun. An average F-measure of about 70% was achieved in these

experiments. The largest training set (for CausFunc0) included 38 verb-noun pairs and all test sets had the size of 15 instances.

Alonso Ramos *et al*. [1] proposed an algorithm for extracting collocations following the pattern *support verb + object* from the FrameNet corpus of examples [22] and checking if they are of the type Oper. This work takes advantage of syntactic, semantic, and collocation annotations in the FrameNet corpus, since some annotations can serve as indicators of a particular LF. The authors tested the proposed algorithm on a set of 208 instances. The algorithm showed an accuracy of 76%. Alonso Ramos *et al*. conclude that extraction and semantic classification of collocations is feasible with semantically annotated corpora. This statement sounds logical because the formalism of lexical function captures the correspondence between the semantic valence of the keyword and the syntactic structure of utterances where the keyword is used in a collocation together with the value of the respective LF.

5 Problem and Research Questions

Wanner in [28] and also we in our previous work [5] interpreted the task of LF detection as a task of classification of verb-noun collocations into two classes: collocations which belong to a particular LF and those which do not belong to this LF. To classify verb-noun collocations, both works applied supervised machine learning methods. In the training set supplied to machine learning algorithms, hypernyms of both verb and noun of each collocation extracted from the Spanish WordNet [25, 27] were used as features. In order to retrieve hypernyms, all words in the training set of verb-noun collocations were annotated with Spanish WordNet senses as well as with their respective LFs.

In the experiments in [28], LFs were detected with an F-measure of about 70% which can be considered sufficiently well however not excellent. The author of [28] analyzes the reasons of classification errors and concludes that two of them are caused by limitations of the Spanish WordNet. Firstly, some senses are absent in this lexical resource: for example, in *observar la costumbre* (lit. observe the custom) *observar* means *follow*, *keep*; however, this sense is absent in the Spanish WordNet. Secondly, some descriptions in the Spanish WordNet are imprecise: for example, semantic descriptions of *periodico* (newspaper) and *libro* (book) differ from each other to a great extent in spite of the

fact that these two words are similar; for a more detailed discussion of imprecise descriptions see [28].

In our experiments reported in [5] we did not include in the dataset those collocations in which the verb and/or the noun do not have their proper senses in the Spanish WordNet. Nevertheless, the performance of supervised classifiers with 10-fold cross validation on the training set did not improve a lot: we obtained an F-measure of about 73%.

It is obvious that some classification errors are due to faults of the supervised learning methods themselves, but we suppose that another obstacle can be found in an insufficient ability of some verb sense definitions to distinguish the semantics of lexical functions, i.e., the meanings of verbs they acquire in collocations. So our hypothesis is that in spite of the fact that such verb sense definitions do represent the meanings of verbs in collocations, the quality of such representation in the part of hypernyms corresponding to such definitions in some cases is not sufficient for discriminating lexical functions.

We mentioned in the Introduction that in a verb-noun collocation, the noun, as the base of the collocation, is used in its typical sense, though the verb, being the collocate and thus semantically dependent on the noun, is not used in its typical meaning but the noun imposes another meaning on the verb. In this work we want to study the correlation between the quality of verb definitions viewed as sets of respective hypernyms and the ability of machine learning methods to discriminate among lexical functions.

Consequently, in this research we intend to respond to the following research questions:

1. To what measurable degree does the meaning of the verb in a collocation differ from the typical meaning of the same verb?
2. Is such degree the same or different for different lexical functions?
3. Is the WordNet sense (represented as a set of hypernyms) which corresponds to the meaning of the verb in a collocation able to distinguish lexical functions and if yes to what degree?
4. How can we measure the correlation between lexical functions and WordNet senses?

To find answers to these questions, we designed three types of experiments with a dataset of Spanish verb-noun collocations annotated with lexical functions [5, 23] and senses of the Spanish WordNet version 2000611 [25, 27]. In the experiments of all types we used supervised

machine learning techniques and hypernyms of both the verb and the noun in a collocation as features, including the verb and the noun themselves as zero-level hypernyms.

1. Experiments on the whole dataset as in our previous work [5]. We repeated these experiments since we use a different number of examples for some lexical functions and a more recent version of Weka thus aiming at a more adequate comparison of the results of these experiments with the results of the other experiments in this work.
2. Experiments on only such collocations of the dataset in which the verb has a sense other than 1. Commonly, a list of senses in WordNet is ordered by frequency, so sense 1 is the most frequent meaning of a word which can be considered as its typical meaning. Thus, the training set in this kind of experiments includes collocations in which the meaning of the verb differs from its typical meaning. (Here we have to remark, that for some collocations, the meaning of the verb in a collocation is most frequently met in corpora and thus is put as sense 1. In such a case, most frequent does not mean most typical. However, what meaning should be considered typical is another research issue; here for our purposes we will adopt the interpretation of typical as most frequent.)
3. Experiments on the same collocations as in the experiments of type 2, but for each verb, we change its sense to sense 1.

To put it simpler in the text that follows, we use Experiment 1, Experiment 2, and Experiment 3 to refer to experiments of type 1, 2, and 3, respectively.

6 EXPERIMENTS

6.1 *Experiment 1*

In Experiment 1 on automatic detection of lexical functions, we used a dataset of Spanish most frequent lexical verb-noun functions [5, 23] compiled by manually annotating each word with the Spanish WordNet [25, 27] senses, and each verb-noun pair as a particular LF or FWC (free word combination). Verb-noun pairs in the dataset are the first 1000 samples in a list of verb-noun pairs retrieved from the Spanish Web

Corpus with 116,900,060 tokens [6, 24] and ordered by frequency. Table 1 presents the statistics of our LF dataset.

Table 1. Lexical functions found in 1000 most frequent verb-noun pairs in the Spanish Web Corpus. For each LF, the number of instances (#) is given as well as their total frequency (Freq) in the corpus; FWC is free word combination (verb-noun pair which is not a collocation)

LF	#	Freq
Oper1	280	165319
FWC	202	70211
CausFunc1	90	45688
CausFunc0	112	40717
Real1	61	19191
Func0	25	17393
IncepOper1	25	11805
Oper2	30	8967
Caus2Func1	16	8242
ContOper1	16	5354
Manif	13	3339
Copul	9	2345
CausPlusFunc0	7	2203
Func1	4	1848
PerfOper1	4	1736
CausPlusFunc1	5	1548
Real2	3	1547
FinOper1	6	1476

LF	#	Freq
PerfFunc0	1	1293
Caus1Oper1	2	1280
Caus1Func1	3	1085
IncepFunc0	3	1052
PermOper1	3	910
CausManifFunc0	2	788
CausMinusFunc0	3	746
Oper3	1	520
LiquFunc0	2	514
IncepReal1	2	437
Real3	1	381
PlusOper1	1	370
CausPerfFunc0	1	290
AntiReal3	1	284
MinusReal1	1	265
AntiPermOper1	1	258
ManifFunc0	1	240
CausMinusFunc1	1	229
FinFunc0	1	178

In Section 2, we did not explain the meaning of all LFs presented in Table 1, only the meaning of the most frequent ones. Definitions and examples for the rest of LFs in Table 1 can be consulted in [8].

An interesting fact can be observed in Table 1: the frequency of verb-noun collocations tagged as Oper1 is higher than the frequency of free verb-noun combinations (FWC). This fact re-affirms the significance of a correct analysis and interpretation of collocations in automatic processing of texts in natural languages.

For our experiments, we chose the first eight LFs in Table 1. Note that FWC stands for free word combinations which are not considered as belonging to lexical functions. The first eight LFs have a sufficient number of samples which allows their usage in supervised machine learning techniques. However, as a training set we used all samples of

the data set only excluding two types of examples. To the first type belong erroneous instances which were retrieved automatically due to parser errors, for example, combinations containing non-letter symbols. The second type of samples which we excluded from the training set are such for whose verb and/or noun the Spanish WordNet does not have an appropriate sense, such deficiency of this widely used dictionary was mentioned in [28].

After removing the latter two types of samples from the dataset, our training set included 900 verb-noun combinations. Table 2 presents the eight LFs we experimented with and their respective number of instances, and Table 3 gives examples of each LF in Table 2.

Table 2. Lexical functions used in Experiment 1

LF	# of instances
Oper1	266
Oper2	28
IncepOper1	24
ContOper1	16
Real1	60
Func0	16
CausFunc0	109
CausFunc1	89
Total	608

We applied supervised machine learning algorithms implemented in Weka 3-6-12-x64 [29, 30] to classify each sample in the training set as belonging to a particular LF or not (binary yes-no classification) using 10-fold cross validation. Each sample was represented as a set of all hypernyms of the verb and all hypernyms of the noun including the verb and the noun as zero-level hypernyms. Hypernyms were retrieved from the Spanish WordNet.

As mentioned in Section 5, such experiments were performed by us in previous work and reported in [5]. However, we considered it necessary to repeat the same experiments, first of all, due to the fact that here we use a more recent version of Weka and, for some LFs, a different number of samples in the training set than in [5]. Secondly, we intend to compare the results of our previous experiments in [5] with Experiments 2 and 3 performed in this research. To make a fair and adequate comparison we will have all the experiments done with the same implementation version of machine learning algorithms and on the

same training set. Finally, in this paper we will give a more extended report of the results of Experiment 1 than for the same type of experimentation performed in [5].

Table 3. Examples of lexical functions from our the training set

LF	Examples of collocations	
	Spanish	English translation
Oper1	*realizar un estudio*	do a study
	cometer un error	make an error
	dar un beso	give a kiss
Oper2	*recibir tratamiento*	receive treatement
	obtener una respuesta	get an answer
	sufrir daño	suffer a damage
IncepOper1	*iniciar un proceso*	begin a process
	tomar la palabra	take the floor
	adoptar la actitud	adopt the attitude
ContOper1	*seguir un curso*	follow a course
	mantener un contacto	keep in touch
	guardar silencio	keep silent
Real1	*satisfacer una necesidad*	satisfy a need
	lograr un objetivo	reach a goal
	resolver un conflicto	resolve a conflict
Func0	*el tiempo pasa*	time flies
	una posibilidad cabe	there is a possibility
	la razón existe	there exists a reason
CausFunc0	*crear una cuenta*	create an account
	formar un grupo	form a group
	hacer ruido	make noise
CausFunc1	*ofrecer una posibilidad*	offer a possibility
	causar un problema	cause a problem
	crear una condición	create a condition

Tables 4–7 present the results of the experiments described in [5] but conducted now as we explained above. In the results, we included the best 10 classifiers in terms of F-measure for each of the eight lexical functions given in Table 2.

The overall average best F-measure for eight lexical functions used in Experiment 1 is 0.734 or about 73%. The work of Wanner [28] reviewed in Section 4 reports an average F-measure of about 70% in two experiments on detection of the following lexical functions: Oper1, Oper2, ContOper1, CausFunc0, Caus2Func1, IncepFunc1, FunFunc1,

Real1, and Real2. Our result of 73% shows a slight improvement compared with [28]; however, such comparison is not fair since we did not experiment with all LFs and the same set of LF instances as in [28].

Table 4. Ten best classifiers on detection of Oper1 and Oper2, respectively

Oper1	
Classifier	F-m
trees.SimpleCart	0.879
rules.PART	0.873
trees.BFTree	0.872
bayes.Bayesian LogisticRegression	0.868
meta.Attribute SelectedClassifier	0.867
meta.Bagging	0.867
trees.LADTree	0.866
meta.END	0.865
meta.FilteredClassifier	0.865
meta.Ordinal ClassClassifier	0.865
Average best	0.869

Oper2	
Classifier	F-m
functions.SimpleLogistic	0.739
meta.LogitBoost	0.739
rules.DecisionTable	0.723
meta.Bagging	0.711
meta.Attribute SelectedClassifier	0.708
meta.END	0.708
meta.FilteredClassifier	0.708
meta.Ordinal ClassClassifier	0.708
trees.J48	0.708
trees.LADTree	0.708
Average best	0.716

Table 5. Ten best classifiers on detection of IncepOper1 and ContOper1, respectively

IncepOper1	
Classifier	F-m
rules.Prism	0.732
trees.FT	0.711
bayes.Bayesian LogisticRegression	0.700
functions.SMO	0.683
misc.VFI	0.682
rules.Nnge	0.682
trees.LADTree	0.682
meta.RandomCommittee	0.667
trees.Id3	0.650
meta.Attribute SelectedClassifier	0.619
Average	0.681

ContOper1	
Classifier	F-m
lazy.LWL	0.800
rules.DecisionTable	0.800
trees.REPTree	0.800
trees.Id3	0.788
meta.Attribute SelectedClassifier	0.774
rules.Ridor	0.774
trees.BFTree	0.774
trees.SimpleCart	0.774
meta.END	0.750
meta.FilteredClassifier	0.750
Average	0.778

Table 6. Ten best classifiers on detection of Real1and Func0, respectively

Real1	
Classifier	F-m
meta.LogitBoost	0.667
meta.Bagging	0.660
trees.BFTree	0.660
functions.SMO	0.649
rules.Jrip	0.647
rules.Nnge	0.635
trees.LADTree	0.634
trees.FT	0.627
bayes.Bayesian LogisticRegression	0.624
trees.REPTree	0.611
Average best	0.641

Func0	
Classifier	F-m
meta.Attribute SelectedClassifier	0.824
rules.Jrip	0.824
trees.ADTree	0.800
meta.END	0.788
meta.FilteredClassifier	0.788
meta.Ordinal ClassClassifier	0.788
rules.PART	0.788
rules.Ridor	0.788
trees.BFTree	0.788
trees.J48	0.788
Average best	0.796

Table 7. Ten best classifiers on detection of CausFunc0 and CausFunc1, respectively

CausFunc0	
Classifier	F-m
rules.Jrip	0.722
trees.LADTree	0.712
trees.SimpleCart	0.710
trees.BFTree	0.705
trees.REPTree	0.704
meta.Bagging	0.679
trees.FT	0.678
functions.SMO	0.676
trees.ADTree	0.670
bayes.Bayesian LogisticRegression	0.664
Average best	0.692

CausFunc1	
Classifier	F-m
meta.RotationForest	0.744
meta.END	0.732
meta.FilteredClassifier	0.732
meta.Ordinal ClassClassifier	0.732
rules.DecisionTable	0.732
trees.J48	0.732
rules.Jrip	0.729
trees.BFTree	0.727
meta.LogitBoost	0.718
trees.LADTree	0.718
Average best	0.730

Another state of the art paper by Alonso Ramos *et al.* [1] surveyed in Section 4 as well reported an accuracy of 76% on extraction of verb-noun collocations of the type Oper from the FrameNet corpus of examples [22]. Here we will mention that our average F-measure on detection of Oper1 and Oper2 is 0.793 or 79%.

6.2 *Experiments 2 and 3*

In the training set for Experiment 2, we included only such collocations of the original dataset (used in Experiment 1) in which the verb has a sense other than 1, that is, the verb has a sense other than its typical sense. In Experiment 3, the number of each verb sense in the training set used in Experiment 2 is substituted by 1. The purpose of this substitution is to compare the performance of classifiers on LF detection using actual (non-typical) verb senses against the performance of the same classifiers on the same collocations using sense 1 (typical) of the verbs.

We believe that comparison of results of these two experiments will shed light on the research questions posed in Section 5.

Table 8 shows, for each lexical function, the total number of instances (i.e., verb-noun collocations), the number of instances in which the verb has sense 1, and the number of collocations in which the verb has sense other than 1, the latter verb-noun pairs were used in Experiments 2 and 3.

Table 8. Lexical functions

LF	Total # of instances (used in Experiment 1)	# of instances with sense 1 of the verb	# of instances with sense ≠ 1 of the verb (used in Experiments 2 and 3)
Oper1	266	112	154
Oper2	28	22	6
CausFunc0	109	29	80
CausFunc1	89	16	73
IncepOper1	24	3	21
ContOper1	16	2	14
Real1	60	44	16
Func0	16	6	10
FWC	196	123	73

The methodology and procedures applied in these experiments are the same as in Experiment 1: in the training set, each verb-noun collocations is represented as a set of hypernyms of the verb and the noun, and the training set was submitted to all applicable to this data type supervised learning methods implemented in Weka 3-6-12-x64 [29, 30].

7 RESULTS AND DISCUSSION

The results of ten best classifiers in Experiment 1 are presented in Section 6.1, see Tables 4-7. However, in this section we refer to the performance of some classifiers in Experiment 1 which did not appear among the best ten ones, but since their performance is high in Experiment 2, we present their values of F-measure in Experiment 1 to make a comparison with their performance in Experiments 2 and 3.

Tables 9–12 show the results of all three experiments. We arranged the results in a way convenient for comparison. The tables include the results for each of the eight lexical functions in the format as follows.

The first column contains ten best classifiers in Experiment 2 ordered by performance on the training set in which the verbs have meanings other than 1, i.e., their actual meaning. The respective F-measure for each classifier is given in the second column entitled $\neq 1$ (Exp.2). The third column entitled 1(Exp.3) contains F-measure of the same classifiers applied to the same training set, but in which each verb is assigned sense 1 (Experiment 3). The fourth column entitled (Exp.2)–(Exp.3) includes the difference between two values: F-measure for the case of the verb sense other than 1 and F-measure for the case of substitution of the actual verb sense with sense 1 (the difference between the results of Experiment 2 and Experiment 3). The fifth column gives the values of F-measure for the classifiers in the first column they reached in Experiment 1, where these classifiers were applied to the original dataset described in Section 6.1. For each of the four columns with the values of F-measure, average is given as well.

Now we will discuss the results of the experiments for each lexical function. It can be observed in Table 9 that Oper1 is detected with almost the same F-measure on the whole dataset and on the set with verb senses other than 1 (0.866 and 0.899, respectively). However, when we substituted verb senses other than 1 with sense 1, the performance became notably worse, with an F-measure of 0.808. This observation suggests that actual verb senses fit well the definition of Oper1, *Agent performs* w_0, where w_0 is the noun in a verb-noun collocation.

For example, consider an Oper1 collocation *realizar*_6 *estudio*_5, lit. realize a study; the numbers here are the Spanish WordNet senses. *Realizar*_6 belongs to the synset {*efectuar*_1, *realizar*_6, *llevar_a_cabo*_5, *hacer*_15}, lit. effect, realize, accomplish, do, and its hypernym is the synset {*actuar*_2, *hacer*_6} (lit. act, do). On the other

hand, *realizar*_1 is in the synset {*causar*_1, *realizar*_1, *crear*_1}, lit. cause, realize, create, which has no hypernym. Clearly, *realizar* in *realizar estudio* does not mean *cause* or *create*, therefore, sense 6 of *realizar* is an adequate correspondence to the meaning of this verb in the collocation under consideration. It seems that for the other Oper1 verbs in the dataset, the situation is the same or very similar.

Table 9. Experimental results for Oper1 and Oper2

Oper1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
trees.SimpleCart	0.900	0.815	0.085	0.879
rules.PART	0.900	0.783	0.117	0.873
trees.LADTree	0.900	0.769	0.131	0.866
meta.END	0.900	0.832	0.068	0.865
meta.FilteredClassifier	0.900	0.832	0.068	0.865
meta.OrdinalClassClassifier	0.900	0.832	0.068	0.865
trees.J48	0.900	0.832	0.068	0.865
rules.Jrip	0.900	0.819	0.081	0.857
trees.BFTree	0.897	0.819	0.078	0.872
trees.REPTree	0.896	0.750	0.146	0.854
Average	0.899	0.808	0.091	0.866

Oper2

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
functions.SimpleLogistic	0.800	0.800	0	0.739
meta.AttributeSelectedClassifier	0.800	0.800	0	0.708
meta.END	0.800	0.800	0	0.708
meta.FilteredClassifier	0.800	0.800	0	0.708
meta.OrdinalClassClassifier	0.800	0.800	0	0.708
rules.DecisionTable	0.800	0.800	0	0.723
rules.Jrip	0.800	0.800	0	0.696
rules.OneR	0.800	0.800	0	0.619
rules.PART	0.800	0.800	0	0.681
trees.BFTree	0.800	0.667	0.133	0.694
Average	0.800	0.787	0.013	0.698

The results for Oper2 in Table 9 are quite different from those for Oper1. While for Oper1 the actual verb senses distinguish well the semantics of this function, it can be observed that Oper2 is distinguished with the same F-measure by the actual verb senses (other than 1) and by

sense 1: nine out of ten best classifiers show the same value of F-measure (0.800) in Experiment 2 and Experiment 3.

Certainly, the results for Oper2 are by no means representative of this whole semantic class of verb-noun collocations since we used a dataset with a very small number of positive examples (6 collocations of Oper2 as positive examples and the rest 872 instances as negative examples). However, we submitted this set to the classifiers in order to make some observations that might sketch lines of future research.

The almost equal performance of the classifiers in Experiment 2 and Experiment 3 on Oper2 detection can be explained by some faults in sense definitions given in the Spanish WordNet. As an example, let us consider *sufrir*_3 *cambio*_3, lit. suffer a change. *Sufrir*_3 belongs to the synset {*soportar*_3, *sufrir*_3}, lit. bear, suffer, and has the following two synsets as hypernyms: {*experimentar*_3}, lit. experience, and {*actuar*_2, *llevar_a_cabo*_3, *hacer*_8}, lit. act, accomplish, do.

On the other hand, *sufrir*_1 belongs to the synset {*aguantar*_4, *tolerar*_1, *sufrir*_1, *soportar*_2}, lit. endure, tolerate, suffer, bear, which has one hypernym {*dejar*_2, *permitir*_2}, lit. allow, permit. Although *experience* is the verb with a clear Oper2 semantics, however, it may be considered too general to classify its hyponym *sufrir* (suffer) as a value of Oper2 for *cambio* (change) as an argument. On the contrary, the verbs *endure*, *tolerate*, *bear*, *allow*, *permit* in combination with the noun *change* have a less general and more specific meaning of *undergo* (*a change*) thus serving as better features for Oper2 detection.

The above example also illustrates the fact that in some cases it is not easy to find the most appropriate word sense for a given lexical function. In Table 9 we see as well that the classifier performance on all samples of Oper2 is worse (F-measure=0.698) than on those samples of Oper2 in which the verb has sense other than 1 (F-measure=0.800). We believe that due to the prevalence of verb sense 1 in the dataset for Oper2 (22 examples with verb sense 1 of total 28 examples, see Table 8), the performance on this dataset is lower.

Similar differences among the results of the three experiments considered for Oper2 in the previous paragraph are also observed for Real1 and Func0, see Table 11.

Table 10 presents the results for IncepOper1. Here the performance of classifiers in terms of F-measure on the dataset with verb senses other than 1 is significantly higher than the classifier performance on the whole dataset (0.812 against 0.644). However, if we substitute sense other

than 1 with sense 1, the performance degrades dramatically (0.812 for verb senses other than 1 versus 0.644 for verb sense 1). It may mean that sense 1 introduces noise into the set of features used for classification, and the other senses communicate the semantics of IncepOper1 more precisely.

Table 10. Experimental results for IncepOper1 and ContOper1

IncepOper1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
trees.Id3	0.900	0.571	0.329	0.650
rules.Prism	0.842	0.583	0.259	0.732
rules.Nnge	0.829	0.516	0.313	0.682
trees.LADTree	0.829	0.629	0.200	0.682
functions.SMO	0.821	0.571	0.250	0.683
functions.Logistic	0.810	0.435	0.375	0.569
meta.MultiClassClassifier	0.810	0.435	0.375	0.567
BayesianLogisticRegression	0.789	0.286	0.503	0.700
functions.SimpleLogistic	0.789	0.429	0.360	0.556
rules.PART	0.703	0.439	0.264	0.615
Average	0.812	0.489	0.323	0.644

ContOper1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
lazy.LWL	0.857	0.880	–0.023	0.800
rules.DecisionTable	0.857	0.923	–0.066	0.800
functions.SimpleLogistic	0.857	0.923	–0.066	0.733
BayesianLogisticRegression	0.857	0.923	–0.066	0.714
rules.Ridor	0.839	0.963	-0.124	0.774
meta.AttributeSelectedClassifier	0.828	0.889	–0.061	0.774
trees.BFTree	0.828	0.889	–0.061	0.774
trees.SimpleCart	0.828	0.923	–0.095	0.774
meta.END	0.828	0.889	–0.061	0.750
meta.FilteredClassifier	0.828	0.889	–0.061	0.750
Average	0.841	0.909	–0.068	0.764

For example, consider the collocation *tomar*_6 *poder*_1, lit. take the power. *Tomar*_6 belongs to the synset {*asumir*_2, *tomar*_6}, lit. assume, take, and has a hypernym synset {*comenzar*_7, *iniciar*_7, *empezar*_6}, lit. commence, initiate, begin. Let us compare the latter with sense 1 of *tomar*: it is in the synset {*conseguir*_1, *tomar*_1, *sacar*_1, *obtener*_1},

lit. get, take, receive, obtain, and has no hypernym, therefore, first, its meaning is most general in this branch of the Spanish WordNet graph, and secondly, its meaning is very different from the semantics of *tomar* in *tomar poder*. The same is true for other IncepOper1 collocations: observe that most of them have verb senses other than 1 (21 out of 24, see Table 8) which represent the IncepOper1 semantics sufficiently well.

Table 11. Experimental results for Real1 and Func0

Real1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
trees.LADTree	0.692	0.667	0.025	0.634
rules.Prism	0.643	0.818	–0.175	0.608
functions.SMO	0.621	0.593	0.028	0.649
rules.Nnge	0.621	0.609	0.012	0.635
trees.FT	0.621	0.720	–0.099	0.627
meta.END	0.609	0.545	0.064	0.606
meta.FilteredClassifier	0.609	0.545	0.064	0.606
meta.OrdinalClassClassifier	0.609	0.545	0.064	0.606
trees.J48	0.609	0.545	0.064	0.606
rules.PART	0.609	0.545	0.064	0.602
Average	0.624	0.613	0.011	0.618

Func0

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
meta.AttributeSelectedClassifier	0.857	0.783	0.074	0.824
rules.Jrip	0.857	0.720	0.137	0.824
trees.ADTree	0.857	0.900	–0.043	0.800
meta.END	0.857	0.526	0.331	0.788
meta.FilteredClassifier	0.857	0.526	0.331	0.788
meta.OrdinalClassClassifier	0.857	0.526	0.331	0.788
rules.PART	0.857	0.750	0.107	0.788
trees.J48	0.857	0.526	0.331	0.788
trees.REPTree	0.857	0.571	0.286	0.788
functions.SMO	0.857	0.857	0	0.778
Average	0.857	0.668	0.189	0.795

Results for ContOper1 in Table 10 are surprising. If the verb senses other than 1 (which pretend to be the actual ones according to a human expert) are changed to sense 1, this improves the classifier performance.

Experiment 2 (verb senses ≠ 1) showed an average F – measure of 0.841, but Experiment 3 (verb senses ≠ 1 substituted with 1) showed an average F-measure of 0.909. The classifier performance on the whole dataset in Experiment 1 is poorer with an average F-measure of only 0.764. It seems that the semantics of ContOper1 *continue to do what is denoted by the noun* is expressed as verbs' typical sense, i.e., sense 1.

Table 12. Experimental results for CausFunc0 and CausFunc1

CausFunc0

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2)| –(Exp.3)	Exp.1
trees.SimpleCart	0.756	0.532	0.224	0.710
trees.LADTree	0.744	0.744	0	0.712
meta.AttributeSelectedClassifier	0.735	0.829	–0.094	0.649
trees.BFTree	0.726	0.818	–0.092	0.705
functions.SimpleLogistic	0.714	0.812	–0.098	0.633
meta.END	0.711	0.829	–0.118	0.628
meta.FilteredClassifier	0.711	0.829	–0.118	0.628
meta.OrdinalClassClassifier	0.711	0.829	–0.118	0.628
trees.J48	0.711	0.769	–0.058	0.628
rules.Jrip	0.704	0.843	–0.139	0.722
Average	0.722	0.783	–0.061	0.664

CausFunc1

Classifier	≠1 (Exp.2)	1 (Exp.3)	(Exp.2) –(Exp.3)	Exp.1
meta.RotationForest	0.771	0.855	–0.153	0.744
trees.BFTree	0.771	0.870	–0.157	0.727
trees.SimpleCart	0.771	0.859	–0.163	0.711
meta.END	0.769	0.861	–0.164	0.732
meta.FilteredClassifier	0.769	0.861	–0.193	0.732
meta.OrdinalClassClassifier	0.769	0.861	–0.167	0.732
trees.J48	0.769	0.861	–0.191	0.732
meta.LogitBoost	0.766	0.892	–0.205	0.718
trees.ADTree	0.766	0.892	–0.254	0.636
meta.AttributeSelectedClassifier	0.762	0.892	–0.145	0.705
Average	0.768	0.870	–0.179	0.717

But what features of sense 1 influence the performance of the classifiers? Let us, as an example, consider *llevar*_5 *vida*_5, lit. spend life. In this collocation, *llevar*_5 does not have synonyms, and its

hypernym is {*usar*_2}, lit. use, so *llevar vida* is interpreted as *use life* in the meaning *continue to live a life*, and this interpretation is correct.

On the other hand, *llevar*_1 belongs to the synset {*acarrear*_1, *traer*_1, *llevar*_1, *transportar*_1}, lit. carry, bring, take, transport, and its hypernym is {*cargar*_1, *transportar*_2, *desplazar*_1, *mover*_1}, lit. bear, transport, displace, move. At the first sight, *move* has nothing to do with the semantics of *spend* in *spend life*. We can note here, that *use* in *use life* implies a process, therefore, continuing to do something; however, *use* fails to serve as an umbrella semantic representation of *continue* for all ContOper1 verbs such as *mantener* (maintain), *seguir* (follow), *guardar* (keep), etc. Since *continue* implies movement or transition from one state to another, such words as *displace*, *move* have a better coverage of ContOper1 verbs and their degree of generalization is sufficient for detecting ContOper1 in verb-noun collocations.

The same phenomenon is observed in detection of CausFunc0 and CausFunc1: the classifier performance is improved if verb senses other than 1 in Experiment 2 are substituted with verb sense 1 in Experiment 3. For CausFunc0, average values of F-measure are 0.722 in Experiment 2 and 0.783 in Experiment 3, and for CausFunc1, 0.768 and 0.870 in Experiments 2 and 3, respectively. Similarly to what was said in the previous paragraph we can say that verbs used in the meaning of CausFunc0 are distinguished better with their typical senses. These are such verbs as *abrir* (open), *agregar* (add), *alcanzar* (reach), *aportar* (contribute), *aprobar* (approve), *causar* (cause), *construir* (construct), *convocar* (call), *crear* (create), *dar* (give), *declarar* (declare), *dejar* (allow), *desarrollar* (develop), *elaborar* (elaborate), *escribir* (write), *establecer* (establish), *formar* (form), *hacer* (do), *introducir* (introduce), *poner* (put), *producir* (produce), *proporcionar* (provide), etc. The same can be said about CausFunc1 verbs: *abrir* (open), *causar* (cause), *constituir* (construct), *crear* (create), *dar* (give), *dejar* (allow), *despertar* (wake), *destacar* (*highlight*), *establecer* (*establish*), *hacer* (do), *ofrecer* (offer), *poner* (put), *prestar* (lend), *producir* (produce), *proporcionar* (provide), *reservar* (reserve). It can be observed from the examples of the verbs, that the same verbs are used in both functions; this explains why the same phenomena of a better performance for verb sense 1 is observed for both functions.

Now let us consider the research questions we posed in Section 5. Firstly, the difference between the classifier performance in

Experiment 2 and Experiment 3 can bear evidence of the degree of similarity / dissimilarity between the typical sense of a verb in a verb-noun collocation and the verb's meaning in the collocation. We observed that such degree of similarity is higher for CausFunc1 (-0.179), ContOper1 (-0.068) and CausFunc0 (-0.061). The lower degree have Real1 (0.011), Oper2 (0.013), Oper1 (0.091), Func0 (0.189), and IncepOper1 (0.323). Secondly, we observed that this similarity degree varies among lexical functions. Thirdly, the results of Experiment 1 and Experiment 2 as well as the similarity degree just mentioned show to what extent the meaning of the verb in a collocation is able to distinguish lexical functions. Lastly, the difference between the classifier performance in Experiment 2 and Experiment 3 can serve also as a measure of correlation between lexical functions and Spanish WordNet senses which also can be used to evaluate the quality of word sense definitions.

8 CONCLUSIONS AND FUTURE WORK

In this work, we have studied to what degree WordNet senses can distinguish among semantic classes of verb-noun collocations represented as lexical functions, a concept of the Meaning-Text Theory by I. Mel'čuk proposed in order to generalize the semantics of restricted lexical co-occurrence or collocations.

We have experimented with supervised machine learning methods on a dataset of Spanish verb-noun collocations annotated with lexical functions and the Spanish WordNet senses. Lexical functions represent such concepts as *do* (what is denoted by the noun), *undergo*, *begin to do*, *continue to do*, etc. Each concept covers a large group of verb-noun collocations thus representing various semantic classes of collocations. Detection of each lexical function was performed as a binary classification using hypernyms of verbs and nouns as features.

We have observed that 5 of 8 lexical functions chosen for the experiments were discriminated well by the actual verb senses with which a human expert annotated them; an average F-measure showed by classifiers on these 5 lexical functions was 0.798. However, 3 of 8 lexical functions were better discerned by classifiers if the actual verb sense was substituted by sense 1, in this case an average F-measure was 0.854 against 0.777 for the case of the actual verb senses of the same lexical functions.

We considered some factors which could cause this phenomenon including imprecise word definitions in WordNet as well as a too high level of generalization of hypernyms. On the other hand, the difference in the performance of classifiers on detection of lexical functions depending on the WordNet senses can be used to measure similarity of senses as well as correlation between semantic classes of verb-noun collocations and WordNet senses; it can also be used to evaluate the quality and discriminative ability of WordNet senses.

In future, we plan to perform a more detailed and extensive analysis of the results obtained in the experiments reported in this work. We also plan to analyze the role of spotting collocations for different text analysis tasks, such as textual entailment [1213], sentiment analysis [16], emotion detection [18, 20], and personality recognition [17], among others.

ACKNOWLEDGEMENTS We are grateful to Vojtěch Kovář and other members of the Sketch Engine team, www.sketchengine.co.uk, for providing us a list of most frequent verb-noun pairs from the Spanish Web Corpus. The work was done under partial support of Mexican Government: SNI, BEIFI-IPN, and SIP-IPN grants 20144534, 20152095.

REFERENCES

1. Alonso Ramos, M., Rambow O., Wanner L.: Using semantically annotated corpora to build collocation resources. Proceedings of LREC, Marrakesh, Morocco, pp. 1154–1158 (2008).
2. Cambria, E., Fu, J., Bisio, F., Poria, S.: AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 508–514 (2015)
3. Cambria, E., Poria, S., Gelbukh, A., Kwok, K.: Sentic API: A common-sense based API for concept-level sentiment analysis. In: #Microposts2014, 4th Workshop on Making Sense of Microposts at WWW 2014, CEUR Workshop Proceedings, vol. 1141, pp. 19–24 (2014)
4. Das, D., Poria, S., Bandyopadhyay, S.: A classifier based approach to emotion lexicon construction. In Natural Language Processing and Information Systems, Springer, pp. 320–326 (2012)
5. Gelbukh, A., Kolesnikova, O.: Supervised learning for semantic classification of Spanish collocations. Advances in Pattern Recognition, Springer Berlin Heidelberg, pp. 362–371 (2010).
6. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The Sketch Engine. Proceedings of EURALEX 2004, pp. 105–116 (2004)

7. Mel'čuk, I. A.: Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In: Wanner, L. (ed) Lexical Functions in Lexicography and Natural Language Processing, Benjamins Academic Publishers, Amsterdam, Philadelphia, PA, pp. 37–102 (1996)
8. Mel'čuk, I. A., Žolkovskij, A. K.: Towards a functioning 'Meaning-Text' model of language. Linguistics, vol. 8(57), pp. 10–47 (1970)
9. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM, vol. 38(11), pp. 39–41 (1995)
10. Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T.: A semantic concordance. In Proceedings of the workshop on Human Language Technology Association for Computational Linguistics, pp. 303–308 (1993)
11. Nakagawa, H., Mori, T.: Automatic term recognition based on statistics of compound nouns and their components. Terminology, vol. 9(2), pp. 201–219 (2003)
12. Pakray, P., Pal, S., Poria, S., Bandyopadhyay, S., Gelbukh, A.: JU_CSE_TAC: Textual entailment recognition system at TAC RTE-6. In System Report, Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook (2010)
13. Pakray, P., Poria, S., Bandyopadhyay, S., Gelbukh, A.: Semantic textual entailment recognition using UNL. Polibits, vol. 43, pp. 23-27 (2011)
14. Poria, S., Agarwal, B., Gelbukh, A., Hussain, A., Howard, N.: Dependency-based semantic parsing for concept-level text analysis. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 8403, Springer, pp. 113–127 (2014)
15. Poria, S., Cambria, E., Ku, L. W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Proceedings of the Second Workshop on Natural Language Processing for Social Media, SocialNLP 2014, pp. 28–37 (2014)
16. Poria, S., Cambria, E., Winterstein, G., Huang, G.B.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems, vol. 69, pp. 45–63 (2014)
17. Poria, S., Gelbukh, A., Agarwal, B., Cambria, E., Howard, N.: Common sense knowledge based personality recognition from text. In: Advances in Soft Computing and Its Applications, MICAI 2013, Lecture Notes in Artificial Intelligence, vol. 8266, Springer, pp. 484-496 (2013)
18. Poria, S., Gelbukh, A., Cambria, E., Das, D., Bandyopadhyay, S.: Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In: Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction, SENTIRE 2012, 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW), IEEE CS Press, pp. 709-716 (2012)

19. Poria, S., Gelbukh, A., Cambria, E., Hussain, A., Huang, G.B.: EmoSenticSpace: A novel framework for affective common-sense reasoning. Knowledge-Based Systems, vol. 69, pp. 108-123 (2014)
20. Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., Bandyopadhyay, S.: Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intelligent Systems, vol. 28, no. 2, 31-38 (2013)
21. Poria, S., Ofek, N., Gelbukh, A., Hussain, A., Rokach, L.: Dependency tree-based rules for concept-level aspect-based sentiment analysis. In: Semantic Web Evaluation Challenge, SemWebEval 2014 at ESWC 2014, Greece, Revised Selected Papers. Communications in Computer and Information Science, vol. 475, Springer, pp. 41-47 (2014)
22. Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice, http://framenet.icsi.berkeley.edu/book/book.pdf/ ICSI Berkeley (2006)
23. Spanish Verb-Noun Lexical Functions, free download at http://148.204.58.221/okolesnikova/index.php?id=lex or http://www.gelbukh.com/lexical-functions
24. Spanish Web Corpus in SketchEngine, http://trac.sketchengine.co.uk/wiki/Corpora/SpanishWebCorpus/
25. Spanish WordNet, http://www.lsi.upc.edu/~nlp/web/index.php?Itemid=57&id=31&option=com_content&task=view
26. Svensson, M. H.: A very complex criterion of fixedness: Non-compositionality. In: Granger, S. Meunier, F. (eds.) Phraseology: an interdisciplinary perspective, Amsterdam & Philadelphia: John Benjamins Publishing Company, pp. 81–93 (2008)
27. Vossen P. (ed): EuroWordNet: A Multilingual Database with Lexical Semantic Networks, Kluwer Academic Publishers, Dordrecht (1998)
28. Wanner, L.: Towards automatic fine-grained classification of verb-noun collocations. Natural Language Engineering, vol. 10(2), pp. 95–143. (2004)
29. Witten, I. H., Frank, E., Hall, M. A.: Data mining: Practical machine learning tools and techniques. Morgan Kaufmann Publishers, MA, USA (2011)
30. The University of Waikato Computer Science Department Machine Learning Group, WEKA download, http://www.cs.waikato.ac.nz/ml/weka/downloading.html

OLGA KOLESNIKOVA
SUPERIOR SCHOOL OF COMPUTER SCIENCES,
INSTITUTO POLITÉCNICO NACIONAL (IPN)
MEXICO CITY, 07738, MEXICO
E-MAIL: <KOLESOLGA@GMAIL.COM >

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 95–113
Received 5/01/2014, accepted 6/02/2014, final 10/12/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Unsupervised Active Learning of CRF Model for Cross-Lingual Information Extraction

MOHAMED FAROUK ABDEL HADY, ABUBAKRELSEDIK KARALI, ESLAM KAMAL, AND RANIA IBRAHIM

Microsoft Research, Egypt

ABSTRACT

Manual annotation of the training data of information extraction models is a time consuming and expensive process but necessary for the building of information extraction systems. Active learning has been proven to be effective in reducing manual annotation efforts for supervised learning tasks where a human judge is asked to annotate the most informative examples with respect to a given model. However, in most cases reliable human judges are not available for all languages. In this paper, we propose a cross-lingual unsupervised active learning paradigm (XLADA) that generates high-quality automatically annotated training data from a word-aligned parallel corpus. To evaluate our paradigm, we applied XLADA on English-French and English-Chinese bilingual corpora then we trained French and Chinese information extraction models. The experimental results show that XLADA can produce effective models without manually-annotated training data.

KEYWORDS: *Information extraction, named entity recognition, cross-lingual domain adaptation, unsupervised active learning.*

1 INTRODUCTION

Named Entity Recognition (NER) is an information extraction task that identifies the names of locations, persons, organizations and other named

entities in text, which plays an important role in many Natural Language Processing (NLP) applications such as information retrieval and machine translation. Numerous supervised machine learning algorithms, such as Maximum Entropy, Hidden Markov Model, and Conditional Random Field (CRF) [1], have been adopted for NER and achieved high accuracy. They usually require large amount of manually annotated training examples. However, it is time-consuming and expensive to obtain labeled data to train supervised models. Moreover, in sequence modeling like NER task, it is more difficult to obtain labeled training data since hand-labeling individual words and word boundaries is really complex and need professional annotators. Hence, the shortage of annotated corpora is the obstacle of supervised learning and limits the further development, especially for languages for which such resources are scarce.

Active learning is the method which, instead of relying on random sampling from the large amount of unlabeled data, reduces the cost of labeling by actively guiding the selection of the most informative training examples: an oracle is asked for labeling the selected sample. There are two settings depending on the oracle type: supervised setting [2], which requires human annotators as oracle for manual annotation, and the unsupervised setting, where the oracle is an automation process. Using different settings, active learning may find much smaller and most informative subset of the unlabeled data pool. The difference between unsupervised active learning and semi-supervised learning [3] is that the former depends on an oracle to automatically annotate the most informative examples with respect to the underlying model. The later depends on the underlying model to automatically annotate some unlabeled data, to alleviate mislabeling noise the model selects the most confident examples.

For language-dependent tasks such as information extraction, to avoid the expensive re-labeling process for each individual language, cross-lingual adaptation, is a special case of domain adaptation, refers to the transfer of classification knowledge from one source language to another target language.

In this paper, we present a framework for incorporating unsupervised active learning in the cross-lingual domain adaptation paradigm (*XLADA*) that learns from labeled data in a source language and unlabeled data in the target language. The motivation of *XLADA* is to collect large-scale training data and to train an information extraction model in a target language without manual annotation but with the help of an effective information extraction system in a source language, bilingual corpus and word-level alignment model.

2 Related Work

2.1 *Cross-lingual Domain Adaptation*

Guo and Xiao [4] developed a transductive subspace representation learning method to address domain adaptation for cross-lingual text classifications. The proposed approach is formulated as a non-negative matrix factorization problem and solved using an iterative optimization procedure. They assume there is a shared latent space over the two domains, such that one common prediction function can be learned from the shared representation for both domains.

Prettenhofer and Stein [5] presented a new approach to cross-lingual domain adaptation that builds on structural correspondence learning theory for domain adaptation. The approach uses unlabeled documents, along with a simple word translation oracle, in order to induce task specific, cross-lingual word correspondences. The analysis reveals quantitative insights about the use of unlabeled data and the complexity of inter language correspondence modeling.

Wan et al. [6] present a transfer learning approach to tackle the cross-lingual domain adaptation. They first align the feature spaces in both domains utilizing some online translation service, which makes the two feature spaces under the same coordinates. They propose an iterative feature and instance weighting (Bi-Weighting) method for domain adaptation. The main idea here is to select features which have distinguished utility for classification from source language and make distributions of source and target languages as similar as possible. In this way, the features useful for classifying instances in source could also be functional for classification on target.

2.2 *Automatic Generation and Annotation of Training Data for Information Extraction*

Yarowsky et al. [7] used word alignment on parallel corpora to induce several text analysis tools from English to other languages for which such resources are scarce. An NE tagger was transferred from English to French and achieved good classification accuracy. However, Chinese NER is more difficult than French and word alignment between Chinese and English is also more complex because of the difference between the two languages.

Some approaches have exploited Wikipedia as external resource to generate NE tagged corpus. Richman and Schone [8] and Nothman et

al. [9] used similar methods to create NE training data. They transformed Wikipedia's links into named entity annotations by classifying the target articles into common entity types. But the article classification seeds also had to be hand-labeled in advance. Kim et al. [10] build on prior work utilizing Wikipedia metadata and show how to effectively combine the weak annotations stemming from Wikipedia metadata with information obtained through English-foreign language parallel Wikipedia sentences. The combination is achieved using a novel semi-CRF model for foreign sentence tagging. The model outperforms both standard annotation projection methods and methods based solely on Wikipedia metadata. *XLADA* does not leverage Wikipedia because its content is poor in some languages like Chinese.

Fu et al. [11] presents an approach to generate large-scale Chinese NER training data from an English-Chinese discourse level aligned parallel corpus. It first employs a high performance NER system on one side of a bilingual corpus. And then, it projects the NE labels to the other side according to the word level alignment. At last, it selects labeled sentences using different strategies and generate an NER training corpus. This approach can be considered as passive domain adaptation while *XLADA* is active learning framework that filters out the auto-labeled data and selects the most informative training sentences.

2.3 *Active Learning for Information Extraction*

Muslea et al. [12] introduced *Co-Testing*, a multi-view active learning framework, where two models are trained on two independent and sufficient sets of features. The most informative sentences are the points of disagreement between the two models that could improve their performance and a human judge is asked for labeling them. On the other hand, *XLADA* looks for the most informative sentences for the target model and we don't have judges.

Jones et al. [3] adapted semi-supervised learning *Co-EM* to information extraction tasks to learn from both labeled and unlabeled data that makes use of two distinct feature sets (training document's noun phrases and context). It is interleaved in the supervised active learning framework *Co-Testing*. *XLADA* differs in that cross-lingual label propagation on a parallel corpus is interleaved for automatic annotation instead of using *Co-EM* approach and that it adopts an unsupervised active learning strategy.

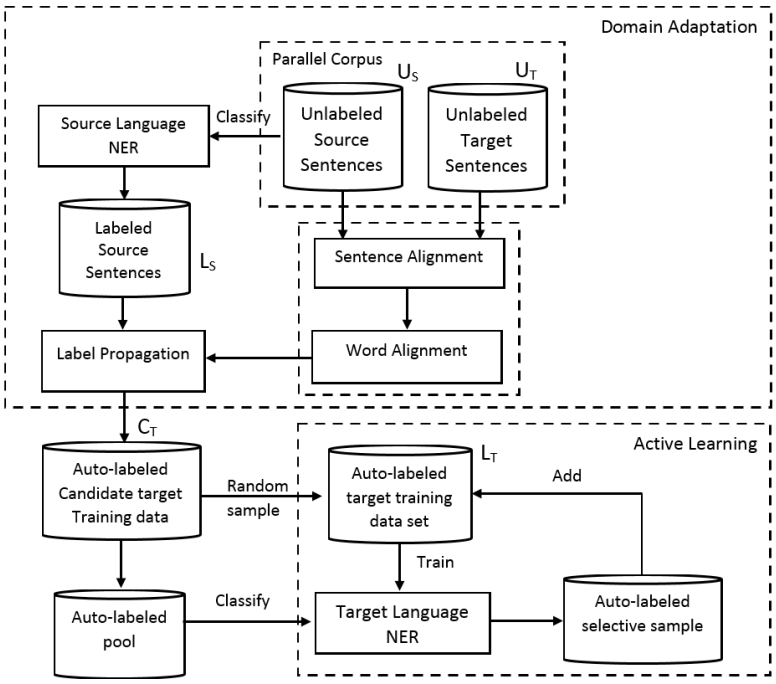


Fig. 1. Architecture of cross-lingual active domain adaptation (*XLADA*)

XLADA is more practical than the framework proposed by Li et al. [13] that depends on cross-lingual features extracted from the word-aligned sentence pair in training the target language CRF model. Hence, it isn't possible to extract named entities from a sentence in the target language unless it is aligned with a sentence in the source language.

3 Algorithmic Overview

The architecture of the proposed combination of cross-lingual domain adaptation and active learning paradigm *XLADA* is shown in Figure 1.

3.1 *Initial Labeling*

Source Language NER An effective source language NER is applied on the source-side of the bilingual corpus U_S to identify named entities

Fig. 2. Projection of named-entity tags from English to Chinese and French sentences

such as person, location, organization names, denote the output L_S. In our experiments, the source language is English and English Stanford NER[1] is used. The system is based on linear chain CRF [1] sequence models that can recognize three types of named entities (Location, Person and Organization).

WORD ALIGNMENT OF PARALLEL CORPUS Sentence alignment and word alignment is performed on the given unlabeled bilingual corpus U_S and U_T. First, sentence level alignment is performed then we applied word dependent transition model based HMM (WDHMM) for word alignment [14].

LABEL PROPAGATION We project the NE labels to the target side of the parallel corpus to automatically annotate target language sentences, according to the result of word alignment, as shown in Figure 2. The output is a set of candidate training sentences. A target sentence is filtered out from the set of candidate training sentences if the number of named entities after label propagation is less than the number of named entities in the source sentence.

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

3.2 *Unsupervised Active Learning*

The amount of auto-labeled sentences in the target language training is too huge to be used for training the information extraction model. Also they are noisy because of the errors in source language NER or word-level alignment. Unsupervised active learning is adopted for selecting high quality training sentences used to train CRF model. The manual annotation of the selected sentences by human judges is replaced with the alignment-based automatic annotation.

We randomly select a set of auto-labeled training sentences L_T. An initial CRF model is trained with L_T. Since a random set of auto-labeled sentences is not sufficient to train a good prediction model in the target language, additional labeled data is required to reach a reasonable prediction model. Afterward, *XLADA* will proceed in an iterative manner.

A pool CP_T of the large amount of auto-labeled sentences is selected. There are two ways to select the sentences in the pool, either a random sample or by assigning a score for each target sentence and finally choose sentences with the highest score (most confident sentences).

The score of each target sentence depends on the score given to its corresponding source sentence in the parallel corpus, as follows:

$$score(S) = \min_{w_i \in S} \max_{c_j \in classes} P(c_j|w_i, \theta_{src})$$

The source NER model θ_{src} assigns probability for each token of how likely it belongs to each entity type: person, location, organization or otherwise. Then, the entity type for each token is the class with maximum probability $P(c_j|w_i, \theta_{src})$. We apply the forward-backward algorithm to compute them.

In each round of active learning, the current target NER model θ_{tgt} tags each target sentence in the auto-labeled pool CP_T. The critical challenge lies in how to select the most informative sentences for labeling. Based on different measurements of target sentence informativeness, we propose the following metric to measure how informative is a given sentence S.

$$inform(S) = \frac{1}{N(S)} \sum_{w_i \in S} \hat{y}(w_i) P(\hat{y}_{tgt}(w_i)|w_i, \theta_{tgt})$$

where $\hat{y}(w_i) = I(\hat{y}_{src}(w_i) \neq \hat{y}_{tgt}(w_i))$ the indicator boolean function between

$$\hat{y}_{src}(w_i) = \arg \max_{c_j \in classes} P(c_j|w_i, \theta_{src})$$

the NE label propagated from the source NER model θ_{src} through alignment information and

$$\hat{y}_{tgt}(w_i) = \arg \max_{c_j \in classes} P(c_j|w_i, \theta_{tgt})$$

the NE tag assigned by the current target NER model θ_{tgt} to the i^{th} word in S.

The most informative sentences are the ones that the target NER model θ_{tgt} didn't learn yet (least confident on its NE prediction) and mismatch with the source NER model θ_{src} where $N(S)$ is the number of tokens in S the two models disagree. Then we select the top N sentences or the ones less than a predefined threshold, add them to L_T with the automatic labels propagated from the source NER model θ_{src} and remove them from the pool CP_T. After new labels being acquired, the target model is retrained on the updated L_T.

3.3 *Conditional Random Field*

Conditional Random Fields (CRFs) [15], similar to the Hidden Markov Models (HMMs) [16] , are a type of statistical modeling method used for labeling or parsing of sequential data, such as natural language text and computer vision. CRF is a discriminative undirected probabilistic graphical model that calculates the conditional probability of output values for a given observation sequence. HMMs made strong independence assumption between observation variables, in order to reduce complexity, which hurts the accuracy of the model while CRF does not make assumptions on the dependencies among observation variables.

Figure 3 shows the graphical representation of liner chain CRFs. Because of its linear structure, linear chain CRF is frequently used in natural language processing to predict sequence of labels Y for a given observation sequence X. The inference of a linear-chain CRF model is that given an observation sequence X, we want to find the most likely sequence of labels Y. The probability of Y given X is calculated as follows:

$$P(Y|X) = \frac{1}{Z(X)} \exp(\sum_{t=1}^{T} \sum_{i=1}^{n} w_i f_i(y_{t-1}, y_t, X, t))$$

where

$$Z(X) = \sum_{Y'} \exp(\sum_{t=1}^{T} \sum_{i=1}^{n} w_i f_i(y'_{t-1}, y'_t, X, t))$$

In the equation, the observation sequence $X = (x_1, \ldots, x_T)$, the label sequence $Y = (y_1, \ldots, y_T)$ where y_t is the label for position t, state feature function is concerned with the entire observation sequence, the transition feature between labels of position $t - 1$ and t on the observation sequence is also considered. Each feature f_i can either be state feature function or transition feature function. The coefficients w_is are the weights of features and can be estimated from training data. $Z(X)$ is a normalization factor.

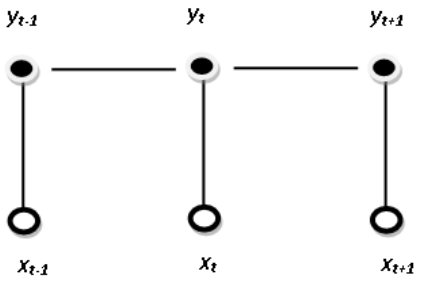


Fig. 3. Graphical representation of linear-chain CRF

4 EXPERIMENTS

4.1 *Datasets*

The performance of *XLADA* is evaluated on the unsupervised learning of Chinese and French NER for named entity recognition of three entity types, person (PER), location (LOC) and organization (ORG). To achieve this goal, unlabeled training data set and labeled test data set is required for each target language. As unlabeled training data, two bilingual parallel corpora is used. The English-Chinese corpus is 20 million parallel sentences and the English-French corpus contains 40 million parallel sentences. The corpora involve a variety of publicly available data sets including United Nations proceedings[2], proceedings of the European

[2] http://catalog.ldc.upenn.edu/LDC94T4A

Parliament[3], Canadian Hansards[4] and web crawled data. Both sides of each corpus were segmented (in Chinese) and tokenized (in English and French).

Table 1. Corpora used for performance evaluation

test set	Chinese	French
#sentences	5,633	9,988
#Person	2,807	3,065
#Location	7,079	3,153
#Organization	3,827	1,935

Table 1 shows a description of the corpora used as labeled test data for *XLADA*. One is the Chinese OntoNotes Release 2.0 corpus [5] and the second is a French corpus manually labeled using crowd sourcing. A group of five human annotators was asked to label each sentence then the majority NE tag is assigned to each token.

4.2 *Setup*

A widely used open-source NER system, Stanford Named Entity Recognizer is employed to detect named entities in the English side of the English-Chinese and English-French parallel corpora.The number of sentences that has at least one named entity detected by the Stanford NER is around 4 million sentences for Chinese and 10 million sentences for French. The features used to train the CRF model are shown in Figure 2. It's worth mentioning that the trainer used here is a local implementation of CRF (not Stanford's implementation) since Stanford's implementation is very slow and memory consuming.

As baselines for comparison, we have studied the following data selection techniques:

- *random sample*: The first NER model was trained on randomly sample of 340,000 and 400,000 sentences from the four million auto-labeled sentences and ten million sentences for Chinese and French language, respectively (upper horizontal dashed lines in Figures 4–6).

[3] http://www.statmt.org/europarl/

[4] http://www.isi.edu/natural-language/download/hansard/

[5] http://catalog.ldc.upenn.edu/LDC2008T04

Table 2. Features used for Named Entity Recognition CRF model

type	extracted features
Shape Features	WordString, WordShape, StartsWithCapital, AllCapital Character N-Grams, Shape Character N-Grams
Brown Clusters [17]	Levels 0, 4, 8, 12
Bags of Words	Date Tokens, Punctuations, Personal Titles, Stop Words
Contextual	Previous 3 words and next 2 word features

– *most confident sample*: the second NER model was trained on the set of the top 340,000 and the top 400,000 most confident sentences (based on the min-max score function defined in Section 3) for Chinese and French, respectively (lower horizontal dashed lines in Figures 4–6).

For active learning, we have randomly chosen 100,000 auto-labeled sentences to train the initial NER for Chinese and French, respectively. And then, we have created a pool (set) of two million sentences where we have two experiments:

– *random pool*: one with a pool of randomly chosen sentences regardless of tagging confidence.
– *most confident pool*: another experiment with a pool of target sentences corresponding to the most confident source sentences selected by min-max score function.

The initial NER is applied on the pool and the informativeness of each sentence is measured using the function defined in section 3.

– *informative addition*: The most informative sentences are the sentences with score less than 0.9. At the end of the first iteration, the labeled training set is augmented with the newly-selected most informative sentences and the target NER is re-trained, this process is repeated for 20 iterations where the final NER for Chinese and French has been trained on 340,000 sentences and 400,000 sentences, respectively.
– *random addition*: another baseline for comparison where in each iteration, a number of auto-labeled sentences in the target language, Chinese or French, is randomly selected, equals to the number of most informative sentences selected in the same iteration at the *informative addition* experiment.

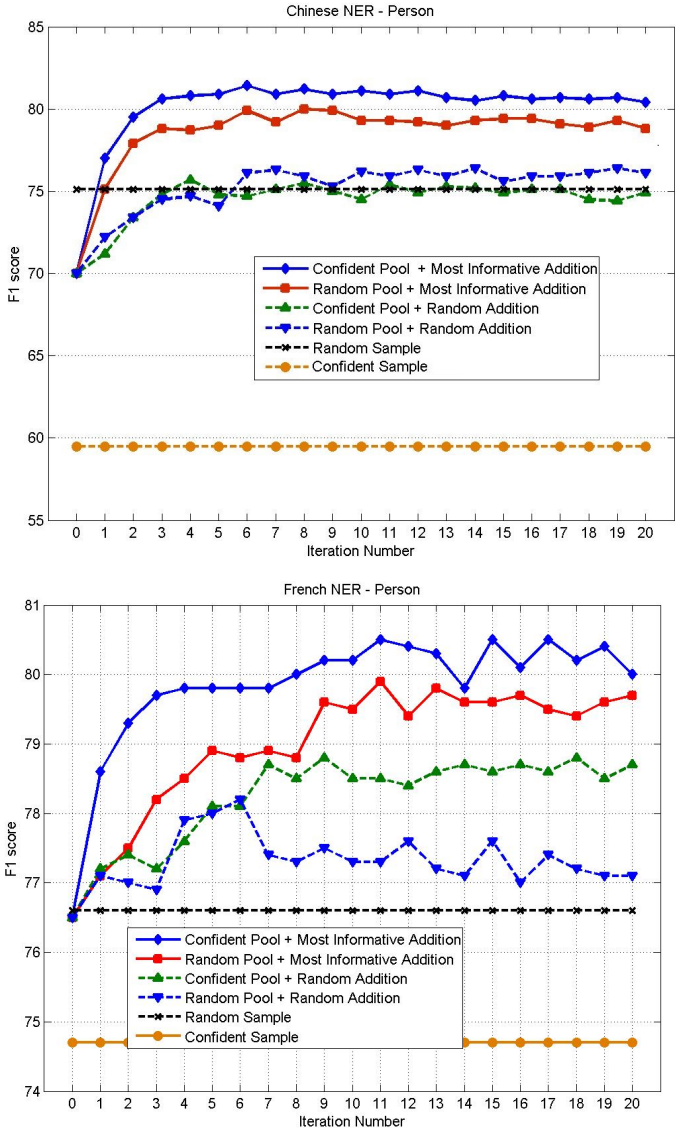


Fig. 4. Performance of unsupervised Chinese and French NER: person

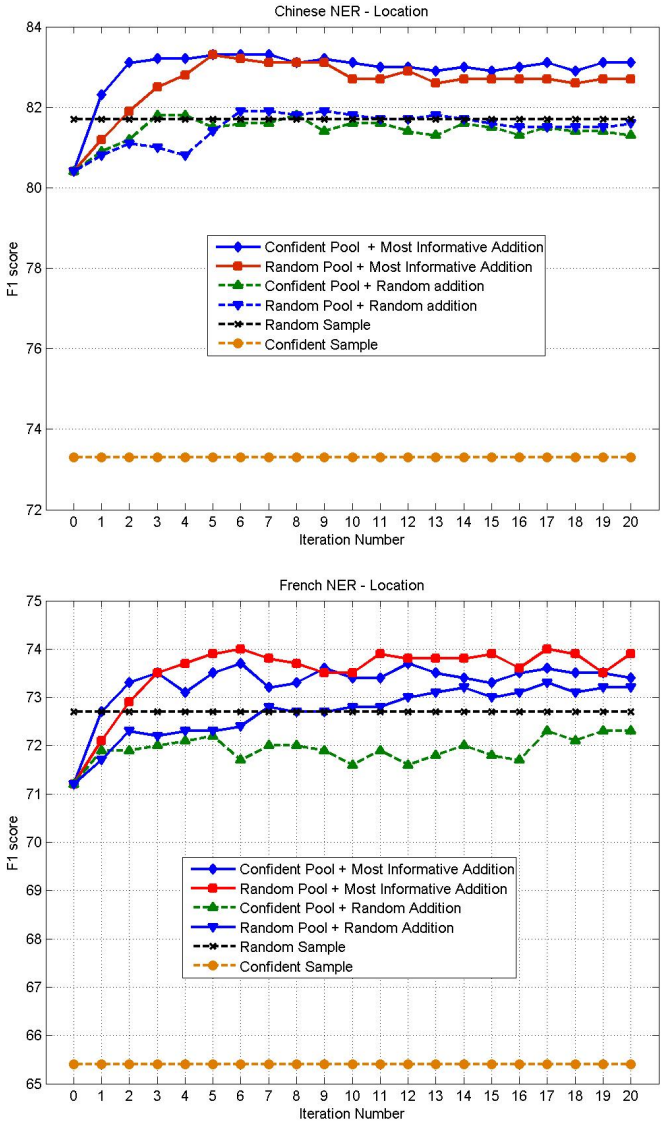


Fig. 5. Performance of unsupervised Chinese and French NER: location

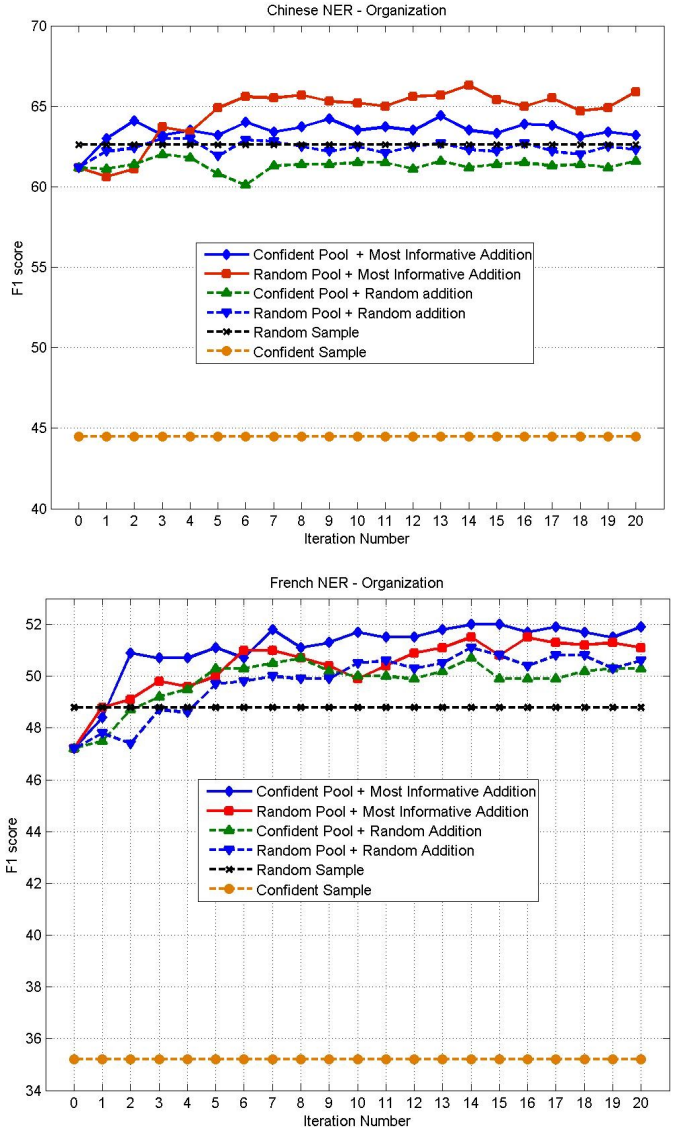


Fig. 6. Performance of unsupervised Chinese and French NER: organization

Table 3. The performance of unsupervised French NER models trained using *XLADA* compared to baselines

	selection method	XLADA		Baseline			
		informative addition		*random addition*		most	random
Entity type	Pool type	*most confident*	*random*	*most confident*	random	confident sample	sample
PER	Precision	78.1	**79.7**	78.6	79.0	77.9	77.0
	Recall	**82.0**	79.6	78.8	75.4	71.9	76.2
	F1	**80.0**	79.7	78.7	77.1	74.7	76.6
LOC	Precision	82.9	82.9	83.2	83.8	73.0	**84.6**
	Recall	65.9	**66.6**	64.0	64.9	59.2	63.8
	F1	73.4	**73.9**	72.3	73.2	65.4	72.7
ORG	Precision	54.1	50.1	52.0	51.7	**59.1**	49.5
	Recall	50.0	**52.2**	48.7	49.6	25.0	48.1
	F1	**51.9**	51.1	50.3	50.6	35.2	48.8

4.3 *Results*

The performance of unsupervised Chinese and French NER systems is reported in Table 4 and Table 3, respectively where the best performing data selection technique is bold faced. Figures 4–6 show the learning curve of target NER models using the different training data selection techniques for Chinese and French, respectively. The F1 measure of both *random sample* NER and *most confident sample* NER is drawn as a horizontal dashed line. The results show that *XLADA* outperforms the random sample baseline.

FOR CHINESE NER For person NE, *XLADA* with *informative addition* using *most confident pool* achieves the highest F1-score 80.4% compared to 59.5% for *most confident sample* and 75.1% for random sample. This is attributed to the increase in person recall from 43.6% and 63.2% to 69.7% and 68.0% respectively. For location NE, *XLADA* with *informative addition* using *most confident pool* achieves the highest F1-score 83.1% compared to 73.3% for *most confident sample* and 81.7% for random sample. This is attributed to the increase in location recall from 64.6% and 74.0% to 76.4% and 75.0% respectively. For organization NE, *XLADA* with *informative addition* using *random pool* achieves the highest F1-score 65.9% compared to 44.5% for *most confident sample* and 62.6% for random sample. This is attributed to the increase in organization recall from 29.4% and 50.3% to 55.2%, respectively.

Table 4. The performance of unsupervised Chinese NER models trained using *XLADA* compared to baselines

	selection method	XLADA		Baseline			
		informative addition		*random addition*		most	random
Entity type	Pool type	*most confident*	*random*	*most confident*	random	confident sample	sample
PER	Precision	**94.9**	93.5	92.9	91.8	93.4	92.4
	Recall	**69.7**	68.0	62.7	65.0	43.6	63.2
	F1	**80.4**	78.8	74.9	76.1	59.5	75.1
LOC	Precision	91.1	**92.2**	90.9	91.1	84.9	91.1
	Recall	**76.4**	75.0	73.6	73.9	64.6	74.0
	F1	**83.1**	82.7	81.3	81.6	73.3	81.7
ORG	Precision	80.8	81.7	87.2	78.6	**91.4**	83.0
	Recall	51.9	**55.2**	47.7	51.6	29.4	50.3
	F1	63.2	**65.9**	61.6	62.3	44.5	62.6

FOR FRENCH NER For person NE, *XLADA* with *informative addition* using *most confident pool* achieves the highest F1-score 80.0% compared to *most confident sample* with 74.7% and *random sample* with 76.6% . This is attributed to the increase in person recall from 71.9% and 76.2% to 82.0%. For location NE, *XLADA* with *informative addition* using *random pool* achieves the highest F1-score 73.9% compared to domain adaptation without active learning: *most confident sample* of 65.4% and *random sample* of 72.7%. This is attributed to the increase in location recall from 59.2% and 63.8% to 66.6%. For organization NE, *XLADA* with *informative addition* using *most confident pool* achieves the highest F1-score 51.9% compared to *most confident sample* of 35.2% and *random sample* of 48.8%. This is attributed to the significant improvement of organization recall from 25.0% and 48.1% to 50.0%.

4.4 *Discussion*

The improvement in recall means increase in the coverage of the trained NER model. This is attributed to the high quality of the training sentences selected by the proposed selective sampling criterion compared to random sampling. In addition, it is better than selecting target sentences where the English NER model is most confident about their corresponding English ones. The reason is that although the English NER model is most confident, this does not alleviate the passive nature of the target NER model as it has no control on the selection of its training data

based on its performance. That is, it implies that the selected sentences do not carry new discriminating information with respect to the target NER model. In all cases, the *random sample* outperforms the *most confident sample*. The reason that selecting only the most confident sentences tends to narrow the coverage of the constructed NER. Figures 4–6 show that *XLADA* achieves the most significant performance improvement in the early iterations, then the learning curve starts to saturate.

In general, the results of *organization* NE type are lower than the results of *Person* and *Location*. The reason is that ORG names are more complex than *Person* and *Location* names. They usually consist of more words, which may result in more word alignment errors and then lead to more training sentences being filtered out. Another reason behind this is that ORG names mostly consist of a combination of common words. Not only for French and Chinese but also English ORG entity recognition is more difficult, which also results in more noise among the ORG training sentences.

5 CONCLUSIONS

The manual annotation of training sentences to build an information extraction system for each language is expensive, error-prone and time consuming. We introduced an unsupervised variant of active learning in the cross-lingual automatic annotation framework that replaces the manual annotation with the alignment-based automatic annotation. It depends on the existence of high quality source language NER model, bilingual parallel corpus and word-level alignment model. A modified score function is proposed as the criterion for selecting the most informative training sentences from the huge amount of automatically annotated sentences. Although the reported results are on the recognition of three entity types, the framework can be generalized to any information extraction task.

REFERENCES

1. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of CoNLL. (2003)
2. Esuli, A., Marcheggiani, D., Sebastiani, F.: Sentence-based active learning strategies for information extraction. In: Proceedings of the 2nd Italian Information Retrieval Workshop (IIR 2010). (2010) 41–45

3. Jones, R., Ghani, R., Mitchell, T., Rilo, E.: Active learning for information extraction with multiple view. In: Proceedings of the European Conference in Machine Learning (ECML 2003). Volume 77. (2003) 257–286
4. Guo, Y., Xiao, M.: Transductive representation learning for cross-lingual text classification. In: Proceedings of IEEE 12th International Conference on Data Mining (ICDM 2012). (2012)
5. Prettenhofer, P., Stein, B.: Cross-lingual adaptation using structural correspondence learning. ACM Transactions on Intelligent Systems and Technology **3(1)** (2012)
6. Wan, C., Pan, R., Li, J.: Bi-weighting domain adaptation for cross-language text classification. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011). (2011)
7. Yarowsky, D., Ngai, G., Wicentowski, R.: Inducing multilingual text analysis tools via robust projection across aligned corpora. In: In Human Language Technology Conference, pages 109-116. (2001)
8. Richman, A.E., Schone, P.: Mining wiki resources for multilingual named entity recognition. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. (2008) 1–9
9. Nothman, J., Curran, J.R., Murphy, T.: Transforming Wikipedia into named entity training data. In: Proceedings of the Australian Language Technology Workshop. (2008) 124–132
10. Kim, S., Toutanova, K., Yu, H.: Multilingual named entity recognition using parallel data and metadata from Wikipedia. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. (2012)
11. Fu, R., Qin, B., Liu, T.: Generating Chinese named entity data from a parallel corpus. In: Proceedings of the 5th International Joint Conference on Natural Language Processing. (2011) 264–272
12. Muslea, I., Minton, S., Knoblock, C.A.: Active learning with multiple views. Journal of Artificial Intelligence Research **27** (2006) 203–233
13. Li, Q., Li, H., Ji, H.: Joint bilingual name tagging for parallel corpora. In: Proceedings of CIKM 2012. (2012)
14. He, X.: Using word-dependent transition models in HMM based word alignment for statistical machine translation. In: Proceedings of the Second Workshop on SMT (WMT), Association for Computational Linguistics. (2007)
15. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of International Conference on Machine Learning (ICML). (2001) 282–289
16. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE. Volume 77. (1989) 257–286
17. Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational Linguistics **18(4)** (1992)

MOHAMED FAROUK ABDEL HADY
MICROSOFT RESEARCH,
CAIRO, EGYPT
E-MAIL: <MOHABDEL@MICROSOFT.COM>

ABUBAKRELSEDIK KARALI
MICROSOFT RESEARCH,
CAIRO, EGYPT

ESLAM KAMAL
MICROSOFT RESEARCH,
CAIRO, EGYPT

RANIA IBRAHIM
MICROSOFT RESEARCH,
CAIRO, EGYPT

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 115–133
Received 17/08/2014, accepted 3/11/2014, final 16/11/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

An Algorithmic Approach for Learning Concept Identification and Relevant Resource Retrieval in Focused Subject Domains

RAJESH PIRYANI, JAGADESHA H., AND VIVEK KUMAR SINGH

South Asian University, India

ABSTRACT

Modern digital world has enormous amount of data on the Web easily accessible anywhere and anytime. This ease of access also creates new paradigms of education and learning. The modern-day learners have access to lot many and in fact one of the best learning materials created in any part of the world. However, despite abundant availability of material, we still lack appropriate systems that can automatically identify learning needs of a user and present them with the most relevant (and best-quality) material to pursue. This paper presents our algorithmic design towards this goal. We propose a text processing-based system that works in three phases: (a) identifying learning needs of a learner; (b) retrieving relevant materials and ranking them; and (c) presenting material to learner and monitoring the learning process. We use know-how of text processing, information retrieval, recommender systems and educational psychology and presents useful and relevant learning material (including slides, videos, articles etc.) to a learner in a focused subject domain. Our initial experiments have produced promising results. We are working towards a Web-scale deployment of the system.

1 INTRODUCTION

With newer form of digital storage devices, large screen readers and fast Internet access, we now have a large volume of anytime anywhere

accessible content. The ease of creation and the resulting rich material is paving the way for new paradigms of education and learning. However, the large amount of online/digital content makes it difficult to identify the most relevant one on a given topic. Imagine, a user, while reading some article/book chapter on 'Introduction to Machine Learning', is automatically presented with related quality resources (such as slides, videos).

This process will not only augment the learning material pursued by the user but will also substantially improve the learning experience/outcome. This automated process of learning resource identification, however, involves complex set of steps. First, we need to know the learning needs of a user, often without an explicit statement by the user. Secondly, good quality and most relevant learning material, in different forms, need to be identified (extracted from the web) and ranked in the order of their relevance and quality. Lastly, selected learning material should be presented to the user and the learning process should be monitored for implicit feedback from the user.

In this paper, we describe our algorithmic design and experimental work towards this theme. We propose to design an adaptable learning resource recommender system, which can effectively enhance the learning outcome by augmenting the learning environment of the user, with additional set of knowledge resources for the given learning concept being pursued by the user. The system assumes that there is a user with a specific learning need. However, the user need not specify it and the system should learn the same through user context and modeling. Thus, when a user is reading a particular piece of a text, the system should automatically extract the learning concepts described in the text, rank them in order of importance and use them as input for additional resource identification.

The additional resource identification process is similar to web search, where relevant articles/slides/videos located anywhere on the web need to be recalled and presented to the user. It is also equally important to measure whether the learning material so recommended is useful and relevant for the user or not. This requires a user interface with capability to monitor and log user learning behaviour (such as user clicks, on screen time etc.). The monitoring provides necessary feedback to the system and allows to adapt to the user learning behaviour and preferences. Thus, the system has three identifiable phases/parts: Concept Identification, Relevant Resource Locator and Adaptable User

Interface. We have used know how from Text Analytics, Computational Linguistics, Information Retrieval, Educational Psychology in designing the system.

The rest of the paper is organized as follows. Section 2 explains the system design and architecture and defines the relevant entities. Section 3 describes the process of parsing the document content, extraction of concepts from different sections, ranking the concepts in the order of their importance. Section 4 explains the learning resource identification and relevance ranking process. Section 5 talks about user modeling and adaptation useful for the system. We present a toy model of the system with the small dataset and experimental results obtained in the focused subject domain in Section 6. The paper concludes with a short discussion and further work to be done for a web-scale deployment of the system. This idea has appeared in a preliminary form in (Singh et al. 2013a) and the part of the work in a different context in (Relan et al. 2013) and (Khurana et al. 2013).

2 System Definition and Architecture

As the first step, an entire system can be depicted by one context diagram, the same is shown in the Figure 1 This figure gives an overview of architecture of the complete system. The system, however, can be more formally described mathematically as follows.

Let

$$U = \{A_1, A_2, \dots, A_n\}, \tag{1}$$

where U is a finite set of attributes$A_1, A_2, \dots, A_n$, which represents user psycho-graphic profile such as on screen time, resources clicked and etc.

$$C = \{c_1, c_2, \dots, c_m\}, \tag{2}$$

where C is a finite set of learning concepts $c_1, c_2, \dots, c_m$ And

$$R = \begin{pmatrix} c_1 \rightarrow r_{11} \dots r_{1j_1} \\ c_2 \rightarrow r_{21} \dots r_{2j_2} \\ \dots \quad \dots \quad \dots \\ c_i \rightarrow r_{i1} \dots r_{ij} \end{pmatrix}, \tag{3}$$

where R is a collection of resources and organized as a linked list where each r_{ij} represents the resource j for the learning concept i, which can be Article, Video, and Slides. Each r_{ij} is sorted according to the ranking

of the resource as explained in the Section 5. Now we introduce a resource match function f which is given as:

$$f: R \rightarrow U \tag{4}$$

which recommends the resources based on the user experience g, and is given by:

$$g = \sum_{i=1}^{m} \sum_{j=1}^{n} h_{c_i r_j} \tag{5}$$

where m is the number of concepts n is the number of resources $h_{c_i r_j}$ represents the j^{th} resource for i^{th} concept h is a resource refining function, which is defined as follows:

$$h_{c_i r_j} = \begin{cases} dfb(c_i r_j) + cf(c_i r_j) * ost(c_i r_j), & cf \text{ and } dfb > 0, \\ dfb(c_i r_j) & if\ cf = 0, \\ 0 & otherwise, \end{cases} \tag{6}$$

where $dfb(c_i r_j)$ is direct feedback from the user for the resource r_j of a particular learning concept c_i. $cf(c_i r_j)$ is click feedback(either 0 or 1) for the resource r_j of a particular learning concept c_i, and $ost(c_i r_j)$ is the on screen time spent on the resource r_j of a particular learning concept c_i.

All these are obtained from user browsing behavior. Our goal is to maximize the function g by refining the recommendations with the most relevant resource for the learning concepts to enhance the understandability.

3 CONCEPT EXTRACTION

The first phase of our system extracts learning concepts from a document being read by the user. This requires a number of tasks as shown in Figure 2 ranging from POS tagging to concept filtering. First of all we parse the textual contents of a document and then use knowledge of linguistics to identify patterns that can represent concepts, there are various methods to do this as described in (Joorabchi and Mahdi 2013). The concepts so identified are subjected to a filtering process for identifying Computer Science (CS) domain concepts. The CS domain

concepts present in a section are then ranked in order of importance for use by the resource retrieval phase. For concept extraction, we had to first do multitude of text extractions from the document, currently we are considering eBook as a document that included extracting Table of Contents, Chapter and Section texts. This was followed by POS tagging and terminological noun phrase identification.

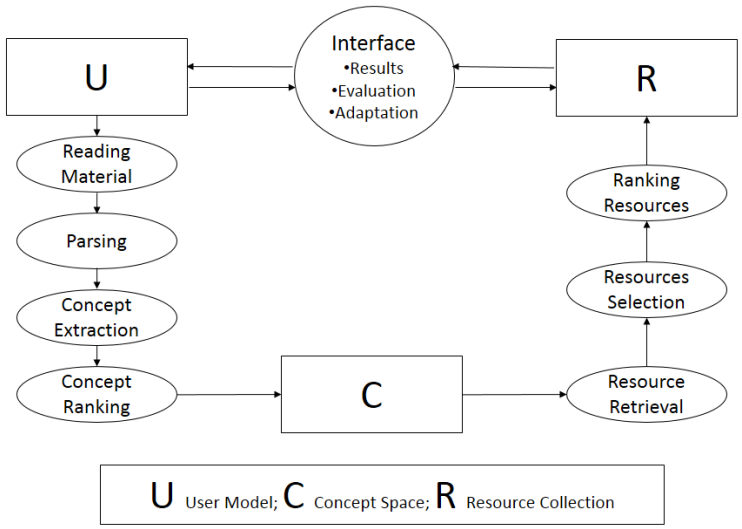


Fig. 1. Architectural Block Diagram of the System

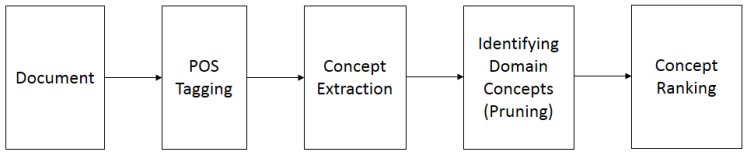


Fig. 2. Concept Extraction Block Diagram

3.1 *Learning Concept Extraction*

We extracted concepts using the terminological noun phrase identification, a set of three kinds of patterns known to represent important noun-phrase based concepts, based on the idea proposed in (Agrawal et al. 2011; Justeson and Katz 1995):

$$P1 = X^*N \quad (7)$$

$$P2 = (X^*NP)?(X^*N) \quad (8)$$

$$P3 = A^*N^+ \quad (9)$$

where, N refers to a noun, P a preposition, A an adjective, and $X = A\ or\ N$. The pattern $P1$ represents a sequence of zero or more adjectives or nouns which ends with a noun. The pattern $P2$ is a relaxation of $P1$ that allows two such patterns separated by a preposition. Examples of the pattern $P1$ may include "probability density function", "fiscal policy", and "thermal energy". Examples of the pattern $P2$ may include "radiation of energy" and "Kingdom of Ashoka". The pattern $P3$ corresponds to a sequence of zero or more adjectives, followed by one or more nouns. In $P3$, an adjective occurring between two nouns is not allowed that means it is a restricted version of $P1$.

It would be pertinent to mention here that symbol $*$ provides for zero or maximal pattern matches and $+$ provides for one or more pattern matches i.e., there is no chance to get "density function" as an extracted pattern if the actual concept mentioned is "probability density function".

Identifying terminological noun phrase patterns from the text require a number of text analytics steps. First of all we have to extract various parts (sections) of the eBook. Then we apply POS tagging on each section extracted. We used Stanford POS tagger[1] for this purpose. This paves the way for identifying terminological noun phrases. The terminological noun phrases so identified are noun phrase based concepts described in a section. A section may contain many such concepts. We have to do two things to proceed further. First, we need to distinguish CS domain concepts from other concepts. Secondly, we need to identify most important learning concepts for a section.

3.2 *Identifying CS Domain Concepts*

The terminological noun phrases extracted represent generic noun-phrase based concepts. Not all of them represent concepts belonging to CS domain. In order to identify relevant R to recommend, we need to know precisely what CS domain learning concepts are described in an eBook section. We have therefore tried to filter out the concepts not in the CS domain. For this, we have used a filtering list containing key

[1] http://nlp.stanford.edu/software/tagger.shtml

learning concepts in CS domain. We understand that this list could not be an exhaustive list of CS domain learning concepts. This may result in losing some CS domain learning concepts, however, the list is appropriate enough to identify key concepts in different subjects of study in CS domain. We have used ACM Computing Curricular Framework document[2] (ACM CCF) as our base CS domain learning concepts. We have augmented these concepts by incorporating in it terms from IEEE Computer Society Taxonomy[3] and ACM Computing Classification System[4]. The augmenting process involved merging the two later documents into the first one, while preserving the 14 categories it is divided into. The combined list is thus a set of 14 different sets of CS domain knowledge areas, each knowledge area containing key concepts (the important ones) worth learning in that area. We use this concepts as our filtering list.

Every concept identified through the terminological noun phrase identification process, is subject to this filtering. However, we cannot do an exact term matching. For example, two terms "algorithm complexity" and "complexity of algorithm" will not be a match, if we go for exact matching scheme. Therefore, we have used Jackard similarity measure, which allows two concept phrases to result in a match even when the word orders in the two are different, or there is an impartial match. The Jackard similarity equation is given in the equation below:

$$Similarity\ (C_R, C_T) = \frac{|C_R \cap C_T|}{|C_R \cup C_T|}, \tag{10}$$

where C_R is the concept in reference document and C_T is a concept in text.

Here, $C_R \cap C_T$ is the set of common words in both concepts, $C_R \cup C_T$ is the set of union of words in both concepts and S stands for the number of elements in the set S. We have to set a threshold value for deciding whether concept C_R and C_T constitute a match. We empirically found a threshold between 0.5 and 0.6, works best for identifying CS domain learning concepts. A simple example could help in understanding the suitability of this threshold. Consider, a concept $C_R =$ *"methods of numerical analysis"* is an identified terminological

[2] http://ai.stanford.edu/users/sahami/CS2013/ironman-draft/cs2013-ironman-v1.0.pdf

[3] http://www.computer.org/portal/web/publications/acmtaxonomy

[4] http://www.acm.org/about/class/2012

noun phrase and a concept $C_T = "numerical\ analysis\ methods"$ is a concept in the CS domain. In this case we get the similarity score = 0.75, greater than threshold and confirming that C_R is a valid CS domain learning concept. Thus, we use the reference list and similarity scores for deciding about every terminological noun phrase extracted from an eBook for being a valid CS domain learning concepts.

3.3 *Ranking Learning Concepts by Importance*

Our implementation tells us that a typical section in an eBook may have occurrences of several valid CS domain learning concepts. Since, we have to recommend resources R for eBook reader pursuing a particular learning concept c_i, we need to select only the most important learning concepts as the input for generating R. This means that if an eBook section results in 10 valid CS domain learning concepts, we can simply not generate R for all the 10 learning concepts, since it would make the R ineffective. We have to, therefore, restrict the learning concepts to be used as input for the process of generating R. This is equivalent to try identifying most important learning concepts in a section. An ideal position will be if we have a scheme to figure out learning concepts semantically, a section is about. But, in the absence of such a scheme to identify semantic tags about learning concepts described in a section, the only option is to use statistical evidence about the concept importance in a section. We have used statistical measures of term occurrence in the concerned section and the entire eBook to rank the learning concepts in order of importance. The rank score (section-rank) of a concept c_i belonging to a particular section S_j is computed as follows:

$$RankScore(c_i, S_j) = Freq(c_i, S_j) + \log\left(\frac{NOLC}{GRank(c_i)}\right) + \alpha \qquad (11)$$

where, $Freq()$ gives the number of occurrences of a particular c_i in a given section, $NOLC$ refers to the total number of CS domain learning concepts extracted from the eBook, $GRank$ is the rank of a c_i in the entire eBook (with highest occurring c_i getting the rank 1) and α is a significance score computed as a weighted sum of metadata, topical terms, wikipedia article, etc, as discussed in the section 3.4.

Thus, we have two ranks for each learning concept, a section-rank and a global-rank. The equation makes it clear that we compute section-rank of a c_i by combining its occurrence measures in the section and the

entire eBook. If the c_i concept refers to the highest ranking concept (rank 1), the $Freq(c_i, S_j)$ value is incremented substantially by addition of log normalized measure of its importance in the entire eBook. On the other hand, if the concept c_i refers to the concept with lowest global rank ($rank = no.of\ concepts$), its log normalized measure value becomes zero (since rank is equal to the number of concepts in eBook) and the section-rank of this concept is only a measure of its occurrence in the concerned section. In this manner, we are able to compute importance of a concept in a given section (measured as section-rank). This is in a sense equivalent to attempting to find the key section (most important) for a learning concept (Agrawal et al. 2010).

3.4 *Computing Significance Score*

When an user is pursuing an article to rank the identified concepts we use significance score which is an weighted sum of wikipedia article, metadata and topical terms i.e, if any concept extracted has an wikipedia article then increase the rank of concept, same way if it has an related concepts mentioned in metadata or topical terms of a document and then increase the rank of the concept so extracted. The mathematical form is as shown below:

$$\alpha = \frac{W + M + T}{3}, \tag{12}$$

where W, M and T are Wikipedia aticle, Metadata, Topical terms respectively and their values are either 0 or 1.

4 RESOURCE IDENTIFICATION AND RANKING

Our resource identification model contains two modules (a) Crawling and (b) Ranking of R. For crawling we have considered a defined set of websites. We use our concept extraction methods to identify the concepts within the link then we associate a tag to the link on the basis of reference library, metadata, co-occurrence and frequency of concepts. For ranking the resources we are invoking web APIs to collect the features of a link such as number of views, comments, likes and rating associated to the link, if no such features are available then we rank on the basis of metadata, wikipedia article and topical terms. The concepts and links are

stored in a database. Once the links are ranked we assign a weightage to the link now this value will vary based on user psycho-graphic profile.

In our existing system, after identifying important learning concepts presented in a section of eBook, we move to second phase of the system, which is to generate recommendations for relevant eResources for the learning concepts being pursued. While a section is being pursued by a reader, we have the key concepts in that section identified and ranked. The top learning concepts then form input for the recommendation generation process. The design of the second part is fairly simple. First of all, we explored about what useful eResources may be readily available. Thereafter, we wrote a JAVA code to invoke search APIs available for this purpose and integrate the results obtained. Our system returns a number of eResources, slides from Slideshare[5] web articles from Google Web Search[6], videos from YouTube[7], microblog posts in the area from Twitter[8], details of professionals working in the area from LinkedIn[9] and related documents from DocStoc[10].

The main objective of designing the recommender system for us was to identify and recommend additional set of eResources for eBook readers. While a reader is reading a particular section of an eBook, we want to provide him with additional learning resources as well as the set of professionals working in that area. While the first is aimed at improving the learning quality and pace; second is to provide an opportunity to the reader to connect to related professionals in the area. For learning concepts pursued by a reader, we generate a set of eResource recommendations. We have designed a web-based interface for this purpose. One important issue is to rank the recommendations based on their relevance to the learning concepts being pursued by the reader. The inherent ranking provided by the APIs invoked is one way to associate relevance to the learning concepts. These APIs use a sophisticated set of algorithms to retrieve only the most relevant results for a search query. We have, therefore, not attempted to rank the retrieved eResources afresh, except while recommending related eBooks

[5] http://www.slideshare.net/about,

[6] http://www.google.com

[7] http://www.youtube.com

[8] http://www.twitter.com

[9] http://www.linkedin.com

[10] http://www.docstoc.com/about/

(where we do rank the recommendations list). Our system design is thus a content-based recommendation system approach (Adomavicius and Tuzhilin 2005; Singh et al. 2011).

5 USER MODELING

User modeling attempts to facilitate the system to improve the quality of R. Our system initially provide the user with most relevant additional R to the user, then our system keeps track of time spent on reading a particular section of an eBook to predict the ability to understand that section. If user spends more time to apprehend the section then we refine the results R with more videos or slides to reduce the comprehension burden.

Our system interact with the user to know more about his/her interests and reconsider the result set R with more related resources of his/her interest. If user click many resource R_{ij} related to a particular c_i then our system revise the results R with more in-depth resources. For example consider user is reading about a concept "machine learning" and the resource set R includes results about "machine learning", "supervised learning" if user clicks several $j^{th}R$ of "supervised learning" then our system will revise the result R with more related resources of "supervised learning".

6 DATASET AND EXPERIMENTAL RESULTS

6.1 *Dataset*

We have performed our experimental evaluation on a moderate sized dataset collected on our own. We collected about 30 eBooks in CS domain from different sources. The text corresponding to various parts of a PDF eBook is extracted using the iText API[11] and programmatically reading the bookmarks. The different parts of an eBook are then parsed at a sentence level, starting with POS tagging and culminating in identification of C (denoted by terminological noun phrases).

[11] http://www.api.itextpdf.com

6.2 *Results*

The JAVA program designed to extract C, their ranks etc. produces a lot of other useful information from eBooks. We have designed an RDF (Resource Description Framework) schema to store the information produced for each eBook. All this information is generated and written automatically (through our program) in the RDF schema. The RDF schema contains rdfs:R for the eBook metadata, C in a section and chapter, concept relations and eBook reviews obtained by crawling the Web. The eBook metadata comprises of eBook title, author, number of chapters, number of pages, eBook price, eBook rating, its main and two related categories as determined from augmented ACM CCF, coverage score, readability score and consolidated sentiment score profile. For each chapter node in the RDF, the entry consists of section and chapter titles, top C with ranks, and relations extracted for the chapter. The populated RDF structure contains a lot of other information for eBooks. We have used only some of this information for our R generation. The other information can be used for a number of purposes like querying about relevant information for the eBook, designing a concept locator in the eBook or designing a semantic annotation environment. A sample example of RDF representation of eBook metadata is as follows:

```
<rdf:RDF
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:book="http://www.textanalytics.in/ebooks/
  Data_Mining_Concepts_and_Techniques_Third_Edition#">
<rdf:Description
rdf:about="http://www.textanalytics.in/ebooks/
Data_Mining_Concepts_and_Techniques_Third_Edition#metadata">
<book:btitle>Data Mining Concepts and Techniques Third
Edition</book:btitle>
<book:author>JiaweiHan,MichelineKamber,Jian Pei
</book:author>
<book:no_of_chapters>13</book:no_of_chapters>
<book:no_of_pages>740</book:no_of_pages>
<book:bconcepts>rule based classification, resolution,
support vector machines,machine learning,...
</book:bconcepts>
<book:main_category>Intelligent Systems</book:main_category>
<book:main_cat_coverage_score>0.051107325
</book:main_cat_coverage_score>
<book:related_category>Programming fundamentals
</book:related_category>
<book:related_category>Information Management
</book:related_category>
```

```
<book:googleRating>User Rating: **** (3 rating(s))
</book:googleRating>
<book:readability_score>56 (Fairly Difficult)
</book:readability_score>
</rdf:Description>
```

In this representation, the category and related category refers to the two closest of the 14 classes defined in ACM CCF. Similarly, other information include readability score, author(s), number of pages etc. The figure 3 shows the RDF Graph for a part of the eBook metadata.

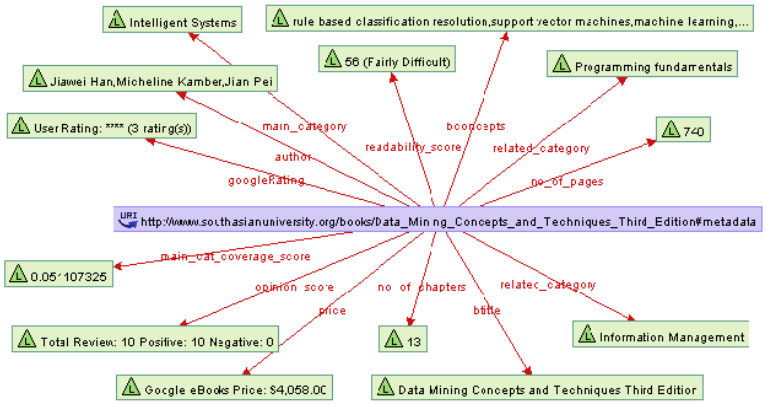


Fig. 3. RDF Graph for Book Metadata

The second key part of the information represented include information about C and their relations in the Chapter node of the RDF schema. A detailed discussion of the RDF schema and relation networks is available in (Uddin et al. 2013).

In the following paragraphs we present snapshot of some results produced at various stages of processing by our system. The snapshot of results shown correspond to a popular eBook on "Data Mining" that describes concepts and techniques of data mining and is a recommended eBook for graduate and research students. During phase 1 of system operation, we extract all probable learning concepts (measured as terminological noun phrases) from a section of the eBook. Then these concepts are filtered using the augmented ACM CCF reference document. For example, from the first chapter of the eBook having title "Introduction", we obtained 1443 concepts before filtering, out of

which 96 concepts refer explicitly to the CS domain. Some example CS domain concepts from beginning portion of this chapter are:

```
business intelligence, knowledge management, entity
relationship, models, information technology, database
management system
```

After obtaining the filtered list of CS domain C in a section of the eBook, we rank them in order of importance. This required that both local (concept occurrence frequencies in the section) and global knowledge (concept ranking for the entire eBook) are available. Thus, we parse the entire dataset of eBooks, identify C in them and rank them in order of importance (assuming whole eBook as unit), beforehand. The concept occurrence frequencies in the currently accessed section are computed at the time of their actual use by the eBook reader. As stated earlier, all the information extracted is also written in an RDF schemea for future retrieval.

The second phase involves generation of R relevant to the most significant C being pursued by the reader. Our R contain eResources of various kinds. The recommendation list R generated by us include videos from YouTube, slides form Slideshare, documents from DocStoc, Web articles from Google Web search, profile ids of professionals working in the area from LinkedIn, Articles or Multimedia from the repository and some others. We present below a sample results for a concept "Data mining" from the first chapter of the eBook used as an example demonstration. An example of recommended videos from YouTube for the concept are as follows:

```
Result for Concept: Data Mining
1.  Thumbnail:
    http://i.ytimg.com/vi/UzxYlbK2c7E/hqdefault.jpg
    URL: http://www.youtube.com/watch?v=UzxYlbK2c7E
2.  Thumbnail:
    http://i.ytimg.com/vi/EUzsy3W4I0g/hqdefault.jpg
    URL: http://www.youtube.com/watch?v=EUzsy3W4I0g
```

An example snapshot of recommended slides from SlideShare for the concept are as follows:

```
Result for Concept: Data Mining
1.  Title:The Secrets of Building Realtime Big Data Systems
    URL:http://www.slideshare.net/nathanmarz/the-secrets-of-
    building-realtime-big-data-systems
2.  Title:Big Data with Not Only SQL
```

```
    URL:http://www.slideshare.net/PhilippeJulio/big-data-
    architecture
```

A sample of recommended documents from DocStoc for the concept is as follows:

```
Result for Concept: Data Mining
1.  Title: Data Mining
    URL: http://www.docstoc.com/docs/10961467/Data-Mining
2.  Title: Data Mining Introduction
    URL: http://www.docstoc.com/docs/10719897/Data-Mining-
    Introduction
```

A snapshot of a part of recommended LinkedIn profiles for the concepts is as follows:

```
Result for Concept: Data Mining
1.  Name: Peter Norvig
    URL: http://www.linkedin.com/in/pnorvig?trk=skills
2.  Name: Daphne Koller
    URL: http://www.linkedin.com/pub/daphne-
    koller/20/3a8/405?trk=skills
```

It would be important to mention here that the results displayed are a very small part of the actual results obtained. More results can be seen at our text analytics portal[12]. Through a similar process of API invocation, we have also generated recommendations for top web links from Google Web Search and top profiles of persons writing on the topic on microblogging site Twitter. We have thus generated recommendations for a comprehensive set of eResources (in addition to identifying the most relevant eBook and its chapter) for a concept being pursued by a learner.

For a given important concept in a section, we also recommend related eBooks (ranked in order of their relevance). The recommended list of related eBooks are at present generated from our dataset collection itself. However, it is not a limitation and we can generate a list of related eBooks (related on the important C under consideration) from the Web. The list of related eBooks is ranked based on a computed sentiment score of their reviews obtained from Google book reviews and from Amazon. It was necessary to rank eBooks since the recommendation list of eBooks is not generated by an API having inherent ranking scheme, but by a concept-bases matching calculation. We want that the most popular

[12] http://www.textanalytics.in

eBooks (measured through wisdom-of-crowds) should be ranked at top and recommended. For this, we have collected user reviews of all the eBooks in the dataset by a selective crawling of Google Book review and Amazon sites. The textual reviews obtained for each eBook are then labeled as 'positive' or 'negative' through a sentiment analysis program designed by us (Singh et al. 2013b, 2013c). Thus for each candidate eBook, we compute sentiment labels and strengths of its reviews (between 10-50 reviews), normalize the strength score (by dividing with number of 'positive' or 'negative' reviews) and use it to rank the eBooks in order of their popularity. Figure 4 shows an example recommendation for the related eBooks recommended for concept “Data Mining”.

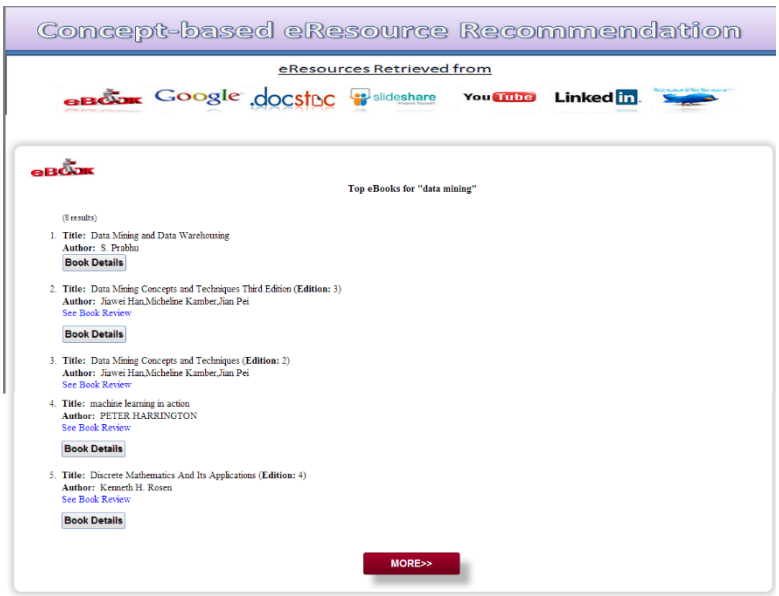


Fig. 4. Recommended eBooks for Concept: Data Mining

7 Conclusion and Future Work

We have presented our experimental work on design of concept-based eResource recommendation system. The system takes as input an eBook being currently read by a user and provides him with additional learning

resources for the learning concepts being pursued by him. The system uses a text analytics approach and works in two phases. In first phase, it identifies the main learning concepts that a user is trying to understand. In the second phase, it generates a set of eResource recommendations that are relevant to the learning concept and provide the user with additional learning material on the concept in concern. The recommender system design proposed and demonstrated by us, appears to be useful for learners.

Evaluation of recommendations is a key parameter of study for recommendation system design. Here, we have used Web APIs for collecting and recommending eResources. These APIs are inherently known to retrieve most relevant results for an information need. There is no such previous system or benchmark against which we can evaluate our system. While the first phase of the system is tested to work appropriately, the results of second phase need some more evaluations for relevance. Our preliminary observation shows that the retrieved and recommended eResources for a learning concept are the most relevant and authoritative ones. We are, however, working towards a wisdom-of-crowd kind of evaluation of the relevance of the recommendation results. Since it is largely a manual effort, it will take some more time to collect user feedbacks from the system hosted on an in-house web portal and being used by volunteers.

There are some possible improvements and extensions of the current work. One of them is to work on a large dataset and explore our system's applicability on open source eBooks from the Web not only for CS but other domains as well. Secondly, we are still working on an appropriate evaluation scheme for ascertaining the quality of recommendations generated. Though, wisdom-of-crowds seem the most natural way, other ways of evaluation may be explored. Thirdly, we wish to extend the system to a full-blown web-based learning resource recommendation system, which can automatically identify users' information needs. Fourth, behavioural and user-based modeling studies may be carried out to evaluate usefulness of the system and to deduce lessons for information need modeling of IR systems. And lastly, the linguistics-based formulations for concept identification, refinement still have possibility of improvement.

ACNKNOWLEDGEMENT This work is partly supported by a UGC-India Research Grant Number 41-642/2012(SR).

REFERENCES

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, doi:10.1109/tkde.2005.99 (2005)
2. Agrawal, R., Gollapudi, S., Kenthapadi, K., Srivastava, N., Velu, R.: Enriching textbooks through data mining. In: Proceedings of the First ACM Symposium on Computing for Development, ACM DEV'10, doi:10.1145/1926180.1926204 (2010)
3. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K.: Data mining for improving textbooks. ACM SIGKDD Explorations Newsletter, vol. 13, no. 2, p. 7, doi:10.1145/2207243.2207246 (2012)
4. Joorabchi, A., Mahdi, A.E.: Automatic keyphrase annotation of scientific documents using Wikipedia and genetic algorithms. Journal of Information Science, vol. 39, no. 3, pp. 410–426, doi:10.1177/0165551512472138 (2013)
5. Justeson, J. S., Katz, S. M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Nat. Lang. Eng., vol. 1, no. 1, doi:10.1017/s1351324900000048 (1995)
6. Khurana, S., Relan, M., Singh, V.K.: A text analytics-based approach to compute coverage, readability and comprehensibility of eBooks. In: 2013 Sixth International Conference on Contemporary Computing (IC3), doi:10.1109/ic3.2013.6612200 (2013)
7. Relan, M., Khurana, S., Singh, V.K.. Qualitative Evaluation and Improvement Suggestions for eBooks using Text Analytics Algorithms. In: Proceedings of Second International Conference on Eco-friendly Computing and Communication Systems, Solan, India (2013)
8. Singh, V. K., Mukherjee, M., Mehta, G. K. Combining a Content Filtering Heuristic and Sentiment Analysis for Movie Recommendations. Computer Networks and Intelligent Computing, pp. 659–664, doi:10.1007/978-3-642-22786-8_83 (2011)
9. Singh, V. K., Piryani, R., Uddin, A., Pinto, D.: A Content-Based eResource Recommender System to Augment eBook-Based Learning. Multi-Disciplinary Trends in Artificial Intelligence, pp. 257–268, doi:10.1007/978-3-642-44949-9_24 (2013)
10. Singh, V. K., Piryani, R., Uddin, A., Waila, P.: Sentiment analysis of Movie reviews and Blog posts. In: Proc. of 2013 3rd IEEE International Advance Computing Conference (IACC), doi:10.1109/iadcc.2013.6514345 (2013)
11. Singh, V. K., Piryani, R., Uddin, A., Waila, P.: Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In: Proc. 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), doi:10.1109/imac4s.2013.6526500 (2013)

12. Uddin, A., Piryani, R., Singh, V.K.: Information and Relation Extraction for Semantic Annotation of eBook Texts. In: Recent Advances in Intelligent Informatics, pp. 215–226, doi:10.1007/978-3-319-01778-5_22 (2014)

RAJESH PIRYANI
DEPARTMENT OF COMPUTER SCIENCE,
SOUTH ASIAN UNIVERSITY, NEW DELHI-110021, INDIA

JAGADESHA H.
DEPARTMENT OF COMPUTER SCIENCE,
SOUTH ASIAN UNIVERSITY, NEW DELHI-110021, INDIA

VIVEK KUMAR SINGH
DEPARTMENT OF COMPUTER SCIENCE,
SOUTH ASIAN UNIVERSITY, NEW DELHI-110021, INDIA

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 135–152
Received 25/06/2014, accepted 22/09/2014, final 12/10/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Implicit Aspect Indicator Extraction for Aspect-based Opinion Mining

IVAN CRUZ,[1] ALEXANDER GELBUKH,[1] AND GRIGORI SIDOROV[1]

Instituto Politécnico Nacional, Mexico

ABSTRACT

Aspect-based opinion mining aims to model relations between the polarity of a document and its opinion targets, or aspects. While explicit aspect extraction has been widely researched, limited work has been done on extracting implicit aspects. An implicit aspect is the opinion target that is not explicitly specified in the text. E.g., the sentence "This camera is sleek and very affordable" gives an opinion on the aspects appearance *and* price*, as suggested by the words "sleek" and "affordable"; we call such words Implicit Aspect Indicators (IAI). In this paper, we propose a novel method for extracting such IAI using Conditional Random Fields and show that our method significantly outperforms existing approaches. As a part of this effort, we developed a corpus for IAI extraction by manually labeling IAI and their corresponding aspects in a well-known opinion-mining corpus. To the best of our knowledge, our corpus is the first publicly available resource that specifies implicit aspects along with their indicators.*

KEYWORDS: *Aspect-based opinion mining, sentiment analysis, conditional random fields.*

1 INTRODUCTION

Opinion mining comprises a set of technologies for extracting and summarizing opinions expressed in web-based user-generated contents. It improves the quality of life for ordinary people by permitting them to consider the collective opinion of other users on a product, political figure,

tourist destination, etc. It improves the incomes of businesses by letting them know what the consumers like and what they do not like. It improves the democracy by permitting political parties and governments evaluate in real time social acceptance of their programs and actions.

Opinion mining depends on accurate detection of opinions expressed in individual documents, such as blog posts, tweets, or user-contributed comments. Such detection can be done at different levels of granularity. For example, the polarity of the whole document can be determined: whether the author expresses a positive or negative opinion. For a comment on a specific product, this level of granularity might be enough. However, it is often desirable to determine sentence per sentence a specific aspect of the product on which opinion is expressed in the given sentence.

Aspect-based Opinion Mining [1, 2] considers relations between the aspects of the object of the opinion and the document polarity (positive or negative feeling expressed in the opinion). Aspect are also called opinion targets. An aspect is a concept on which the author expresses their opinion in the document. Consider, for example, a sentence "The optics of this camera is very good and the battery life is excellent." We can say that the polarity of this review of a photo camera is positive. However, more specifically, what the author likes are *optics* and *battery life* of this camera. These concepts are the aspects of this opinion.

Aspect Extraction is the task of identifying the aspects, or opinion targets, or a given opinionated document. The aspects can be of two types: explicit aspects and implicit aspects. Explicit aspects correspond to specific words in the document: in our example, the opinion targets *optics* and *battery life* explicitly appear in the document. In contrast, an implicit aspect is not specified explicitly in the document. Consider the sentence "This phone is inexpensive and beautiful." This sentence expresses a positive opinion on *price* and *appearance* of the *phone*. These aspects would be explicit in an equivalent sentence "The price of this phone is low and its appearance is beautiful."

While there are many works devoted to the explicit aspect extraction, implicit aspect extraction is much less studied. Implicit aspect extraction is much more complicated than explicit aspect extraction. However, implicit aspects are ubiquitous in the documents, as the following example from the corpus described in [1] shows: *This is the best phone one could have. It has all the features one would need in a cellphone: It is lightweight, sleek and attractive. I found it very user-friendly and easy to manipulate; very convenient to scroll in menu etc.* In this example,

the expressions "lightweight," "sleek" and "attractive," "user-friendly" and "to scroll in menu," and "easy to manipulate" correspond to the aspects *weight*, *appearance*, *interface*, and *functionality* of the phone, correspondingly. The latter expression can be also interpreted as referring, more specifically, to the aspect *menu* of the phone. While these concepts are not explicitly mentioned in the text, they are introduced implicitly by the words that are present. We call such words, which are clues to infer the implict aspects of the opinion, *Implicit Aspect Indicators* (IAI).

Note that in this paper, we do not consider any noun as an aspect; instead, we assume that there is a pre-defined set of *aspects* (variables) of which IAI indicate the *values*. IAI differ from *implicit aspect expressions* defined by Liu [3] as "aspect expressions that are not nouns or noun phrases" in that IAI semantically refer to the values of the pre-defined aspects, irrespectively of their own surface part of speech; below we give examples of IAI expressed by nouns and noun phrases; see also Table 3.

The task of identification of implicit aspects, or implicit aspect extraction, is usually done in two phases. First, the IAI are identified in the document, e.g., "user-friendly." Next, they are mapped to the corresponding aspects, e.g., *interface*. In this paper, we concentrate on the first step: identification of the IAI, a task that we call *implicit aspect indicator extraction*, or IAI extraction. Existing approaches to the second step (mapping IAI to aspects) are mentioned in Section 2.

An IAI could be a single word, such as "sleek," a compound, such as "user-friendly," or even a complete phrase, such as "to scroll in menu" in the above example.

IAI can be of different parts of speech: in "This MP3 player is really expensive," the IAI "expensive" suggesting the aspect *price* is an adjective; in "This camera looks great," the IAI "look" suggesting *appearance* is a verb; in "I hate this phone. It only lasted less than six months!," the IAI "lasted" suggesting *durability* of the phone is a verb.

The following examples shows IAI as nouns or noun phrases: in "Even if I had paid full price I would have considered this phone a good deal" the IAI "good deal" suggest the aspect *price*; in "Not to mention the sleekness of this phone" the IAI "sleekness" suggest the aspect *appereance*; in "The player keeps giving random errors" the IAI "random errors" suggest the aspect *quality*; in "This phone is a piece of crap" the IAI "piece of crap" suggest the aspect *quality*.

Different IAI can correspond to the same implicit aspect. Such IAI can refer to different values of this aspect, e.g., "beautiful" or "ugly" for *appearance*, or to the same value, in which case they can be approxi-

mately synonymous, e.g., “beautiful,” “pleasant,” or “sleek,” or participate in approximately synonymous expressions, e.g., “it is pleasant to look at this phone” or “the designer showed a very good taste.”

Many authors consider only polarity or sentiment words as possible IAI. For instance, in the sentence “this phone is beautiful” the word “beautiful” has positive polarity, so it is natural to assume that indicates an opinion about some aspect, which in this case is *appearance*. Note that here the assumption is that both the aspect and the value are expressed cumulatively by the same word. While such an approach works in many cases, it fails in other cases. For example, in the sentence “the designers of this camera did a very good job,” the word “designers” is not a sentiment word, but still implies the aspect *appearance*, which is not be implied by the only polarity word “good” in this sentence. Namely, here the implicit aspect is indicated by one word and its value by another word. The IAI and the word that gives its value can even appear in different sentences, e.g., “I love this phone. It works even in areas with very low signal,” where “love” gives the value of the aspect *reception*.

It is not always trivial to decide whether a value of an aspect implies positive or negative opinion. For example, “the phone is very heavy” vs. “the battery lasts a lot”: it is common sense that high weight for a phone is bad and high capacity for a battery is good. This is called *desirable facts*: even if the text does not contain an explicit opinion about the aspect to be good or bad but only communicates an objective fact about it, the fact should still be desirable, which implies a positive opinion, or undesirable, which implies a negative opinion. Another example: “The phone has the latest version of Android” is an objective fact, and there are no opinion words in this text; however, for a phone to have the latest version of the operating system is desirable and thus the opinion implied by it about the aspect *operating system* is positive.

In this paper, we present a novel method for IAI extraction. We use a supervised learning approach, based on sequential labeling with Conditional Random Fields (CRF). Our results show that our approach outperforms existing approaches.

To the best of our knowledge, there is no corpus for the IAI extraction task. Thus we developed such a corpus. For this, we manually labeled the IAI and their corresponding aspects in a well-known corpus for opinion mining [1]. The corpus is publicly available for research purposes.[1]

[1] Available on www.gelbukh.com/resources/implicit-aspect-extraction-corpus, visited on November 10, 2014.

The paper is organized as follows. Section 2 discusses the related work. Section 3 presents the scheme and the features we used. Section 4 describes our experimental methodology. The results are given in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Hu and Liu [1] were the first to introduce the notion of aspect extraction in the context of opinion mining, as well as to differentiate the explicit and implicit aspects. In their work, however, they addressed only explicit aspects (using statistical rules) and did not consider any treatment of implicit aspects. Later, Popescu and Etzioni [4] and by Blair-Goldensonh [5] further improved their method.

Currently, there exist a number of methods for aspect extraction. In this paper we will present a method based on a supervised learning technique. Thus, in the rest of this section we will focus on the supervised learning methods.

The task of aspect extraction is a particular case of information extraction task. There exist various methods for the latter task [6, 7], of which the most dominant ones are based on *sequential labeling*. There are two main techniques for sequential labeling: Hidden Markov Models (HMM) and Conditional Random Fields [8] (CRF).

Various methods have been applied for aspect extraction. Lexicalized HMM were applied to extract the opinions paired with the corresponding explicit aspects [9]. CRF were used by various authors for explicit aspect extraction [10–13].

Fewer works addressed implicit aspect extraction. The first system of this kind, OPINE [4], was introduced in order to achieve better polarity classification. Unfortunately, this system is not well-documented and not available for public.

All methods for implicit aspect extraction we are aware of rely on what we in this paper call IAI. In all works, only sentiment words are considered as candidates for IAI. Clustering was used to convert such IAI into explicit aspects, basing on the statistics of co-occurrence of explicit aspects and sentiment words in the sentences [14]. Two-phase co-occurrence association rule mining was used to relate implicit and explicit aspects [15]. In another rule-based method, explicit aspects were identified in the text and then implicit aspects were mapped to them by clustering the pairs of explicit aspects and sentiment words that were candidates to implicit aspects [16].

Recently, rule-based frameworks has shown very promising results for extraction of implicit and explicit aspects [17] and for aspect-based sentiment analysis [18, 19], especially by using concepts and not single words [20]. Identification of sentiment words and sentiment orientation of the text is in turn a task that has been dealt with using rule-based approaches [21], machine-learning methods [22–24], and lexical resources [25, 26].

3 METHODOLOGY

In what follows we describe the scheme used for IAI extraction and the features we used during our experiments.

3.1 *IAI Extraction*

The objective is to label words from an opinionated input text as IAI. Figure 1 shows an example. There is an opinionated sentence as input. The output is a set of duples. Each duple consists of a token of the sentence and the label of a class assigned by an IAI extraction method. The label 'I' is for the class "IAI" and the label 'O' is for the class "Other." The words "sleek" and "affordable" are classified as IAI.

INPUT: "*This phone is sleek and very affordable.*"
OUTPUT: {('*This*',O),('*phone*',O),('*is*',O),
('*sleek*',I),('*and*',O),('*very*',O),
('*affordable*',I),('.',O)}

Fig. 1. IAI extraction example

We cast the task of IAI extraction as a sequence labelling task. Let $X = \{x_1, ..., x_m\}$ be a set of observations and $Y = \{y_1, ..., y_m\}$ a set of assigned labels to those observations. The objective is to predict the set of labels $Y' = \{y_{m+1}, ..., y_n\}$ given a set of new inputs $X' = \{x_{m+1}, ..., x_n\}$ with a model obtained with the observed data X and the given labels Y. The sequential labelling method used is Conditional Random Fields.

3.2 *Conditional Random Fields*

Conditional Random Fields (CRF) is a probabilistic graphical framework for building probabilistic models to segment and label sequences of data. It takes a discriminative approach. More generally, a CRF is a log-linear model that defines a probability distribution over sequences of data given a particular observation sequence. Lafferty et al. [8] defined a CRF on a set of observations X and a set of label sequences Y as follows: Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v\epsilon V}$ so that Y is indexed by the vertices of G, then (X, Y) is a conditional random field in case, when conditioned on X the random variables Y_v, obey the Markov property with respect to the graph:

$$p(Y_v|X, Y_u, u \neq v) = p(Y_v|X, Y_u, u \sim v), \tag{1}$$

where $u \sim v$ means that w and v are neighbors in G. This property describes the fact that the conditional probability of a label Y_v depends only on a label Y_u iff there is *affinity* with Y_v, i.e. $(Y_v, Y_u)\epsilon E$.

The joint distribution over the label sequences Y given X has the form:

$$p_\theta(y|x) \propto exp\left(\sum_{e\epsilon E,k} \lambda_k f_k(y|_e, x) + \sum_{v\epsilon V,k} \mu_k g_k(v, y|_v, x)\right), \tag{2}$$

where x is the data sequence, y is a label sequence, $y|_S$ is the set of components of y associated with the vertices in subgraph S, f_k and g_k are *feature functions* and θ is the set weight parameters

$$\theta = (\lambda_1, \lambda_2, \lambda_3, ...; \mu_1, \mu_2, \mu_3, ...).$$

The feature functions f_k and g_k are a set of functions that maps a set of observations X to a real number, typically to the subset $\{0, 1\}$. These functions are built in order to model an observation X_i as a vector. We assume that the features are given and fixed. They are usually Boolean and crafted by hand. For example, a vertex feature f_k can be true (i.e, f_k maps the observation X_i to 1) if the word X_i is upper case and the tag Y_i is proper noun.

For our proposed approach we used a particular case of this framework: Linear Chain Conditional Random Fields Sequence Labeling [8]. This is a supervised method for predicting label sequences given a set of

observations. The implementation used in our experiment was the CRFClassifier included in the Stanford NER[2] [27]. This classifier is a Java implementation of arbitrary-order linear-chain CRF sequence models.

3.3 *Features*

The data used for training the CRF-based labeller was taken from the dataset described in Section 4.1. We pre-processed the data by removing punctuation and stop words. Capitalized and upper-case words were left as they are.

The Stanford NER includes a Java class named NERFeatureFactory. This class implements several feature extraction methods. One can enable (with a configuration file) specific feature extractors to use them with the CRFClassifier in order to build a feature vector. We used this class to built such feature vectors for our experiments.

Given a sequence of words, we construct a feature vector for each word to be labelled. These feature vectors contains the following features encoded:

1. Word Features: These are features that indicate which word type is the actual instance to be labelled.
2. Character n-grams features: These are features that indicate if a substring appears in a word. These type of features have been proved useful in Name Entity Recognition tasks [28]. The substrings are from the corpus types. A restriction on these n-grams is that they are not be larger than 6 characters. This restriction is because with larger n-grams the training becomes very expensive in terms of computational power, with little classification performance gain. Other restriction is that they do not contain either the beginning or end of the word. We determined experimentally that n-grams with these properties give better performance.
3. Part of Speech (POS) tag features: The POS tag of the word. For these features, one must provide the POS tag for each token in a sentence as input. We used the NLTK POS Tagger[3] for tagging.
4. Context Features: These are the word, tag and the combination word-POS tag of the previous and next word of the current instance to be labelled.

[2] http://nlp.stanford.edu/software/CRF-NER.shtml
[3] http://nltk.org/

5. Class sequences features: These are the combination of the given particular word with the labels given to the previous words. We used a label window of 2, i.e. the labels of the 2 previous words plus the current word and as features.
6. Word bi-gram features.

4 Experimental Setup

The general description of our experimental setup is as follows: first we developed a corpus for IAI extraction, then we defined the different metrics and validation methods for our experiments. Finally, we defined the baselines to compare the performance of our approach.

4.1 *Dataset*

We noticed that there was no suitable dataset for our experiments. As explained in Section 1, limited work has been done in extracting implicit aspects. Moreover, the task that we call IAI extraction was not defined since the common approach to infer implicit aspects was to take sentiment words as the best words to infer such aspects. Therefore, it is natural that there are no resources for IAI extraction (as far as we know). As a result of this, we developed the first corpus for IAI extraction.

Hu and Liu [1] developed a corpus for explicit aspect extraction. This corpus has been widely used in many opinion mining subtasks. We used the texts of this corpus to create a new one for IAI extraction. We labelled the text indicating the IAI and their corresponding implicit aspects. We only selected sentences that have at least one implicit aspect in order to label the corpus. Therefore, we did not label every opinionated sentence.

Table 1 shows some of the properties of the IAI corpus. It consists of 314 Amazon reviews of 5 products in the electronics commodities domain: a DVD player (the column "DVD" in the table), a Canon camera ("Canon"), an MP3 player ("MP3"), a Nikon camera ("Nikon") and a Nokia Cellphone ("Phone"). This table describes the number of reviews per document. It also describes how many words and sentences a review has on average.

The corpus statistical properties at different granularity levels are shown in Table 2. The the name of each column is the same as the name of the columns in Table 1. This table is divided in 3 section for each granularity level:

– Sentence level.
– Token level.
– Type level.

The sentence-level section shows how many sentences are in the document, how many sentences of the documents have at least one IAI (shown in the row labelled as "IAI#") and the percentage of sentences that have at least one IAI ("IAI%"). The token-level and type-level section describes the same properties for these granularity levels.

Table 1. Corpus Properties.

	DVD	Canon	MP3	Nikon	Phone
Reviews	99	45	95	34	41
Words per Review	572.3	1236.4	1575.5	924	1085.3
Sentences per Review	7.47	13.26	18.90	3.64	13.31

Table 2. Statistical Properties.

	DVD	Canon	MP3	Nikon	Phone
Sentence level					
Sentences	740	597	1796	346	546
IAI#	147	63	155	36	44
IAI%	19.86%	10.55%	9.03%	10.40%	8.05%
Token level					
Tokens	56661	55638	149676	31416	44497
IAI#	164	79	214	50	66
IAI%	0.289%	0.141%	0.142%	0.159%	0.148%
Type level					
Types	1767	1881	3143	1285	1619
IAI#	72	63	136	40	42
IAI%	4.07%	3.34%	4.32%	3.11%	2.59%

The POS distribution for the IAI labeled in the corpus is shown in Table 3. Each row represents a general Penn Treebank POS tag. The first row represents all the tags that are adjectives (JJ, JJR, JJS), the second one represents the noun tags (NN,NNS, NNP, NNPS) and the third one represents the verb tags (VB, VBD, VBG, VBN, VBP, VBZ). The last

row is the rest of tags seen with an IAI. The IAI column shows how many vocabulary words were seen labeled with the given tag. The third column describes how many words with the given tag were seen in sentences with IAI. The fourth column shows the tag distribution observed in the IAI.

Table 3. Corpus POS distribution

POS	IAI	POS in IAI Sentence	P(IAI)
JJ	157	527	0.2818
NN	167	1692	0.3000
VB	220	1112	0.3836
other	19	3900	0.0346

4.2 *Metrics and Validation Methods*

We used our annotated corpus as gold standard. The labeled IAI include compounds and phrases. Labeled words as IAI must match those labeled as IAI in the corpus, which are counted as true positives (tp). Those words that do not match are counted as false positives (fp). False negatives (fn) are words labeled as IAI in the corpus that were not extracted as IAI.

We measured the precision and recall. Precision P and recall R are defined as

$$P = \frac{tp}{tp + fp}, \qquad R = \frac{tp}{tp + fn}.$$

The performance metric used was the $F1$ Score. It is defined as

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}.$$

The results were obtained in a 10-fold cross validation setup.

4.3 *Baseline Approach*

The first baseline is to label the sentiment words as IAI for each sentence in a review [14–16]. We call this baseline BSLN1. We use the sentiment lexicon used in [2] to determine the opinion polarity of words. This lexicon is conformed by two word lists. The first list is conformed

by ”positive words”, which are words that suggest a positive opinion in an opinionated context (e.g ”awesome”). The second list is conformed by ”negative words”. These words suggest a negative opinion (e.g. ”awful”).

The algorithm for this baseline is as follows: for each word in a sentence we determine if this word is in any of the two list of the lexicon. If it is, we label it as IAI.

We propose a second baseline based on text classification, which we call BSLN2: we implemented a Naive Bayes (NB) text classifier. The classifier was trained with the texts of our developed corpus. The task of this classifier is to determine whether a sentence has at least one IAI or not. If a sentence is classified as a one with IAI, we label the sentiment words as IAI.

The features used in the NB classifier were:

- Corpus vocabulary stems. We exclude stop words.
- The best 500 bi-gram collocations obtained by a Point-wise Mutual Information association measure.

Finally, we also implemented a second-order Hidden Markov Model sequence labeller. This is the standard method for sequence labelling. We called this method BSLN3. We trained this labeller with our corpus. Since the labeller is a second order HMM, we use bigrams and trigrams as features. The training data is pre-processed as follows:

- The words that appear fewer than 5 times in the corpus (rare words) are changed in the training data for the label *RARE*.
- The rare words that contain at least one numeric character are changed for the label *NUMERIC*
- The rare words that consist entirely of capitalized letters are changed for the label *ALLCAPS*

All baselines were implemented in Python. We used the NB Classifier included in NLTK for the BSLN2.

5 RESULTS

Table 4 shows the performance of BSLN2 classifying sentences with at least an IAI within.

Table 5 compares the performance of the baselines and our CRF-based approach with different features combinations. We call *WT* the

Table 4. BSLN2 sentences classification performance

	Precision	Recall	F1-Score
Extraction of Sentence with IAI	0.25	0.37	0.30

combination of the word and tag features (points 1 and 3 from the features description in Section 3.3). *CNG* features are the character n-grams features (point 2). *CNTX* are context features and word bigram features (points 4 and 6). *CLS* are the class sequence features (point 5).

We observed that the WT features give the greatest precision. However the recall is poor.

The CNG features give a recall boost. These features capture the morphological properties of the words (roots, prefixes, suffixes). Words with similar morphological properties tend to be semantically similar. For example the sentence "This phone looks great" could be rephrased as "The phone's look is great" or even "This phone looked great with its case." The root "look" present in the previous sentences is the best IAI to infer the *appearance* aspect. Therefore words with this root should have greater probability of being extracted as IAI. The drawback is that these features decrement the overall precision because more words that are not IAI but contain these character n-grams will have greater probability of being extracted.

The CNTX and CLS features improve both precision and recall. The best performance is obtained with the combination of WT, CNG, CNTX and CLS features.

Table 5. IAI Extraction performance with different features

	Precision	Recall	F1 Score
BSLN1	0.0381	**0.3158**	0.0681
BSLN2	0.1016	0.1379	0.1170
BSLN3	0.5307	0.1439	0.2264
WT	**0.6271**	0.0575	0.1053
CNG	0.4765	0.1925	0.2742
CNTX	0.5030	0.1148	0.1869
WT,CNG	0.4697	0.1992	0.2795
WT,CNG,CNTX	0.5209	0.2031	0.2932
WT,CNG,CNTX,CLS	0.5458	0.2064	**0.2970**

Our approach is better for this task. It outperforms significantly both BSLN1 and BSLN2. The precision is very high. However the recall is lower than BSLN1.

In order to tradeoff precision for recall in our CRF-based approach, we used a biased CRF classifier [29]. This method allows to set a bias towards the different classes. These biases (which internally are treated as feature weights in the log-linear model underpinning the CRF classifier) can take any real value. As the bias of a class *A* tends to plus infinity, the classifier will only predict *A* labels, and as it tends towards minus infinity, it will never predict *A* labels. These biases are used to manually adjust the precision-recall tradeoff.

We experimented with the *IAI* class bias. We changed the value of this bias within a range of 1.5 to 3.5. We keep the *Other* class bias value fixed to 1. The set of features used are those shown in the last row of Table 5. Table 6 shows the precision, recall and F1 Score of several experiments with different IAI class bias values. Figure 2 shows a graphic of this data.

Table 6. Precision, Recall and F1 Score with different IAI Class bias

IAI Class Bias	Precision	Recall	F1 Score
1.5	**0.5252**	0.2602	0.3479
2.0	0.4636	0.3095	0.3711
2.5	0.4201	0.3503	**0.3820**
3.0	0.3656	0.3850	0.3750
3.5	0.3184	**0.4203**	0.3623

The best F1 Score is obtained with an IAI class bias value of 2.5. It gives a boost of 28.61% in terms of the IAI extraction performance without IAI class bias. Furthermore both precision and recall are higher than any of the baselines.

6 CONCLUSIONS

We have described a model for extracting what we call Implicit Aspects Indicators, which are words that infer implicit aspects of an opinionated document using Conditional Random Fields. We developed a dataset for this task based on a well-know corpus for opinion mining. Also we presented a comparative performance evaluation of our approach with three baselines. The results shown that our approach outperforms significantly

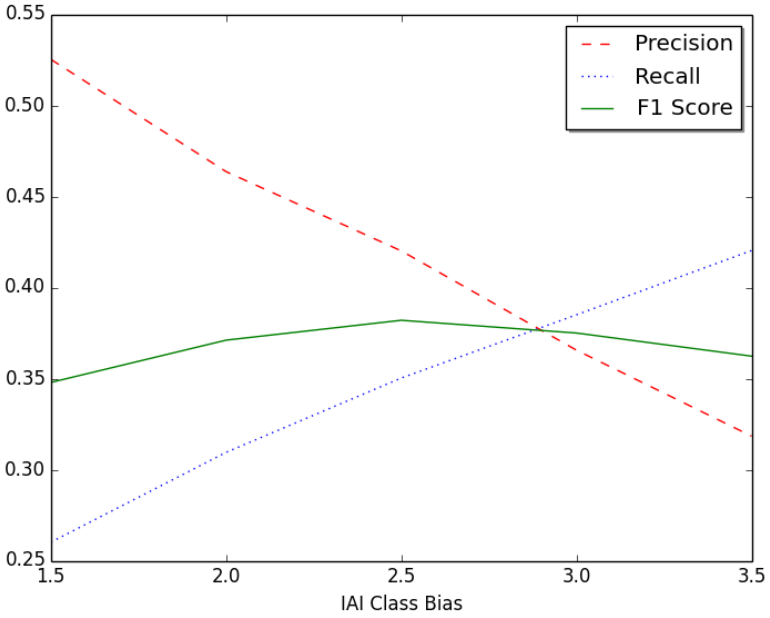


Fig. 2. Precision, Recall and F1 Score with Biased CRF

these baselines. The features used were described and we shown they are not complicated yet quite effective in IAI extraction.

For future work we are going to study new features for this task. We believe that syntactic dependency features could improve the performance [30, 31]. Finally we are working on a Implicit Aspects extraction model based on IAI. We will explore several approaches for mapping IAI with implicit aspects using semantic similarity [32–35].

REFERENCES

1. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. (Aug 2004) 168–177
2. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of First ACM International Conference on Web Search and Data Mining (WSDM-2008), Stanford University, Stanford, California, USA (Feb 2008) 231–240

3. Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers (2012)
4. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2005). (2005) 3–28
5. Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G.A., Reynar, J.: Building a sentiment summarizer for local service reviews. In: Proceedings of WWW-2008 Workshop on NLP in the Information Explosion Era. (2008) 14
6. Hobbs, J.R., Riloff, E.: Information extraction. In Indurkhya, N., Damerau, F., eds.: Handbook of Natural Language Processing. Chapman & Hall/CRC Press (2010) 511–532
7. Mooney, R.J., Bunescu, R.: Mining knowledge from text using information extraction. ACM SIGKDD Explorations Newsletter **7(1)** (2005) 3–10
8. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann Publishers (2001) 282–289
9. Jin, W., Ho, H.H.: A novel lexicalized HMM-based learning framework for web opinion mining. In: Proceedings of International Conference on Machine Learning (ICML-2009). (2009) 465–472
10. Niklas, J., Gurevych, I.: Extracting opinion targets in a single and cross-domain setting with conditional random fields. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010). (2010) 1035–1045
11. Li, F., Han, C., Huang, M., Zhu, X., Xia, Y.J., Zhang, S., Yu, H.: Structure-aware review mining and summarization. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING-2010). (2010) 653–661
12. Choi, Y., Cardie, C.: Hierarchical sequential learning for extracting opinions and their attributes. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010). (2010) 268–274
13. Huang, S., Liu, X., Peng, X., Niu, Z.: Fine-grained product features extraction and categorization in reviews opinion mining. In: Proceedings of the IEEE 12th International Conference on Data Mining Workshops. (2012) 680–686
14. Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Swen, B., Su, Z.: Hidden sentiment association in Chinese web opinion mining. In: Proceedings of International Conference on World Wide Web (WWW-2008). (2008) 959–968
15. Zhen, H., Chang, K., Kim, J.: Implicit feature identification via co-occurrence association rule mining. In: Computational Linguistics and Intelligent Text Processing. 12th International Conference, CICLing 2011, Tokyo, Japan, February 20–26, 2011. Proceedings, Part I. Volume 6608 of Lecture Notes in Computer Science. (2011) 393–404

16. Zeng, L., Li, F.: A classification-based approach for implicit feature identification. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. 12th China National Conference, CCL 2013 and First International Symposium, NLP-NABD 2013, Suzhou, China, October 10–12, 2013, Proceedings. Volume 8202 of Lecture Notes in Computer Science. (2013) 190–202
17. Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP) (2014) 28–37
18. Poria, S., Ofek, N., Gelbukh, A., Hussain, A., Rokach, L.: Dependency tree-based rules for concept-level aspect-based sentiment analysis. In: Semantic Web Evaluation Challenge. Springer International Publishing (2014) 41–47
19. Poria, S., Gelbukh, A., Agarwal, B., Cambria, E., Howard, N.: Sentic demo: A hybrid concept-level aspect-based sentiment analysis toolkit. ESWC 2014 (2014)
20. Cambria, E., Poria, S., Gelbukh, A., Kwok, K.: A common-sense based API for concept-level sentiment analysis. Making Sense of Microposts **1**(1) (2014) 1–2
21. Poria, S., Cambria, E., Winterstein, G., Huang, G.B.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems **69** (2014) 45–63
22. Poria, S., Gelbukh, A., Cambria, E., Das, D., Bandyopadhyay, S.: Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In: Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on, IEEE (2012) 709–716
23. Poria, S., Gelbukh, A., Das, D., Bandyopadhyay, S.: Fuzzy clustering for semi-supervised learning—case study: Construction of an emotion lexicon. Proceedings of MICAI (2012)
24. Das, D., Poria, S., Bandyopadhyay, S.: A classifier based approach to emotion lexicon construction. In: Natural Language Processing and Information Systems. Springer Berlin Heidelberg (2012) 320–326
25. Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., Bandyopadhyay, S.: Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intelligent Systems (2) (2013) 31–38
26. Poria, S., Gelbukh, A., Cambria, E., Yang, P., Hussain, A., Durrani, T.S.: Merging SenticNet and WordNet-Affect emotion lists for sentiment analysis. In: Signal Processing (ICSP), 2012 IEEE 11th International Conference on. Volume 2., IEEE (2012) 1251–1255
27. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005). (2005) 363–370

28. Klein, D., Smarr, J., Nguyen, H., Manning, C.: Named entity recognition with character-level models. In: Proceedings of CoNLL-2003. (2003) 180–183
29. Minkov, E., Wang, R.C., Tomasic, A., Cohen, W.W.: NER systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. (2006) 93–06
30. Sidorov, G.: N-gramas sintácticos no-continuos. Polibits **48** (2013) 69–78
31. Sidorov, G.: Should syntactic n-grams contain names of syntactic relations? International Journal of Computational Linguistics and Applications **5**(2) (2014) 17–38
32. Das, N., Ghosh, S., Gonçalves, T., Quaresma, P.: Comparison of different graph distance metrics for semantic text based classification. Polibits **49** (2014) 51–57
33. Huynh, D., Tran, D., Ma, W., Sharma, D.: Semantic similarity measure using relational and latent topic features. International Journal of Computational Linguistics and Applications **5**(1) (2014) 11–25
34. Wolf, L., Hanani, Y., Bar, K., Dershowitz, N.: Joint word2vec networks for bilingual semantic representations. International Journal of Computational Linguistics and Applications **5**(1) (2014) 27–42
35. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas **18**(3) (2014) 491–504

IVAN CRUZ
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN,
INSTITUTO POLITÉCNICO NACIONAL,
AV. JUAN DE DIOS BÁTIZ, S/N, 07738, MEXICO CITY, MEXICO
E-MAIL: <ICRUZ_B12@SAGITARIO.CIC.IPN.MX>

ALEXANDER GELBUKH
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN,
INSTITUTO POLITÉCNICO NACIONAL,
AV. JUAN DE DIOS BÁTIZ, S/N, 07738, MEXICO CITY, MEXICO
E-MAIL: <WWW.GELBUKH.COM>

GRIGORI SIDOROV
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN,
INSTITUTO POLITÉCNICO NACIONAL,
AV. JUAN DE DIOS BÁTIZ, S/N, 07738, MEXICO CITY, MEXICO
E-MAIL: <SIDOROV@CIC.IPN.MX>

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 153–164
Received 17/08/2014, accepted 4/11/2014, final 14/11/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Evaluating Prosodic Characteristics for Vietnamese Airport Announcements

LAM-QUAN TRAN,[1] ANH-TUAN DINH,[2]
DANG-HUNG PHAN,[2] AND TAT-THANG VU[2]

[1] *Vietnam Aviation Institute, Vietnam*
[2] *Institute of Information Technology, Vietnam*

ABSTRACT

In most languages, the quality of a speech synthesis system relates directly to the diversity of language domain. Each domain, such as sports, entertainments, etc., has its specific grammar structures. The grammar structure plays as an important role for analyzing the prosodic information of utterances in each domain. In this research, we will analyze characteristics of prosodic information of airport domains in Vietnamese and detect most important characteristics related to the sentiment of Vietnamese Airport announcements.

1 INTRODUCTION

Advantages of the text-to-speech system have been utilized for many areas. To build a smooth voice, there are several statistical methods that have been widely researched. In statistical methods, Hidden Markov Model-based speech synthesis provides many benefits to build a high quality TTS system. With HMM, the TTS process is created by two main process: training data and synthesizing the input text achieved from users [3]. With HMM training, the voice can be created with small footprint of sound data [6], with lower than one hour of sound record. Moreover, based on the statistical method, HMM can model the co-articulation between consecutive sound units to provide smooth synthetic

voice. However, one of the main disadvantages of HMM based speech synthesis is the naturalness. HMM training steps tend to neutralize the parameter of the synthesized output sound, including the F0, pitch and duration. The neutralization brings drawback that the voice is over-smooth and has low naturalness.

To overcome the low quality naturalness problem for HMM based TTS, the decision tree component has been improved for its task to detect phonemic and prosodic characteristics of the set of phonemes obtained from the input text [2]. Structure of the decision tree of HMM-based is language dependent, it depends on the grammar structure and prosodic information of input text. Due to that, the decision tree is also domain dependent. Its structure verified in each domains, such as sport, science, etc… However, in general Vietnamese Speech Synthesis System, the structure of decision tree lacks of prosodic information embedded in input text, brings the result that the voice output quality is still average in naturalness.

The main approach to produce a high quality decision tree for every language is to analyze characteristics of input text for a specific domain. In this research, we focus on analyzing information about the input text provided by a set of airport announcements [7]. Based on the analyzing, we detected characteristics provided by rules about prosodic, including part of speech, stress, and intonation of sample airport announcements. The result of this thesis is embedded to enhance the quality of the auto-announcement system of Vietnam Airline [7]. This paper includes three sections: prosodic analysis in airline announcements, improvement in HMM-based speech synthesis system and experiments.

2 Prosody Analysis in Airline Announcements

To understand the prosody phenomenon, the part of speech arrangements in the sentences are demonstrated. By observing F0 contour of training sentences, the relations between POS and Stress, POS and Intonation are established.

2.1 *Stress*

Stress is how a phoneme is underscored in a syllable. In languages such as Russian, English and French, stress is very important. However, in

Vietnamese and other tonal languages, the stress's role is less important than those of Russian, English… In Airline statement, a phoneme can be emphasized with long duration and loud voice. In training corpus, the syllable "*Nam*" in "*Việt Nam*", "*bay*" in "*chuyến bay*", "*đi*", "*số*" in "*quầy số*" or "*cửa số*" and "*VN*" is strengthen and has long durations. In addition, the noun such as space names and names of the flight are also underscored. The long duration appears when the announcers try to emphasize the important information followed the above words. The speakers are demanded to read the statement in a noisy and crowded environment like the airport and they have to ensure the important information can come to the customers.

According to Doan Thien Thuan [1], a syllable's structure can be demostrated as in Table 1. Vietnamese is a tonal monosyllable language, each syllable may be considered as a combination of Initial, Final and Tone components as Table 1. The Initial component is always a consonant, or it may be omitted in some syllables (or seen as zero Initial). There are 21 Initials and 155 Final components in Vietnamese. The total of distinct pronounceable syllables in Vietnamese is 18958 [9] but the used syllables in practice are only around 7000 different syllables [1]. The Final can be decomposed into Onset, Nucleus and Coda. The Onset and Coda are optional and may not exist in a syllable. The Nucleus consists of a vowel or a diphthong, and the Coda is a consonant or a semi-vowel. There are 1 Onset, 16 Nuclei and 8 Codas in Vietnamese. By observation, the stress in the Airline speech utterances lies on Nucleus.

Table 1. Structure of Vietnamese syllable

Tone			
[Initial]	Final		
	[Onset]	Nucleus	[Coda]

2.2 *Intonation*

In our work, ToBI is used to transcribe the intonation in the training sentences. ToBI is a widely used transcription standards. It has been applied in prosodic analysis in many languages such as English, French, and Chinese… The primitive elements in ToBI are low (L) and high (H) tones. The melody of a sentence is divided into many elements. The

elements are classified into two main groups: Phrase-final intonation and Pitch Accent.

Phrase-final intonation is the variation of spoken pitch at the end of a intonation phrase [2]. A sentence can have more than one intonation phrase. The L-L% (low-low) and L-H% (low-high) tags are used in transcription. L-L% is used at the end of a $_{\text{statement}}$ phrases and L-H% is used to mark the end of an emotional phrases and question utterances.

Pitch Accent is the failing and rising trend of pitch contour [2]. The failing trend is described by the H+L* (high-low) tag and L* (low) tag. L+H* (low-high) and H* (high) present the rising trend in the baseline of F0 contour. The following sentence in training set shows an example of using ToBI in Intonation Transcription. F0 contour of the sentence are partly shown in the Figure 1.

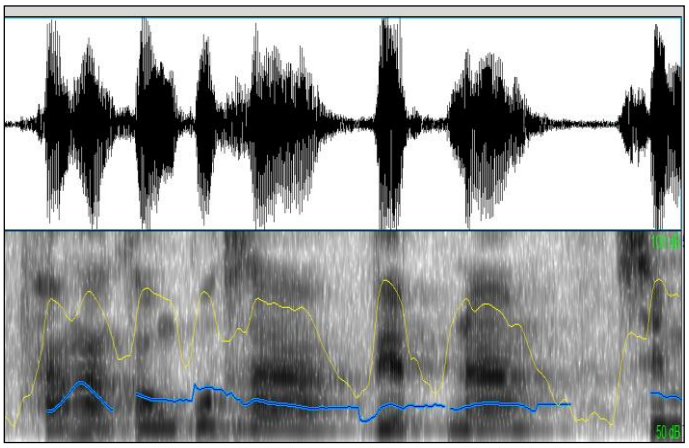


Fig. 1. Spectrogram and F0 contour of the sentence: "*Hãng hàng không quốc gia Việt Nam xin mời hành khách trên chuyến bay VN27 tới cửa số 08 để khởi hành.*" ("Vietnam Airline invites passengers on the flight VN27 please go to board 08 to start")

The relation between intonation transcription and F0 contour is very complicate. Some common relations are described in Table 2.

By observing the set of Airline announcement, the transition network of Vietnamese phrasal melodies can be established as in Figure 3.

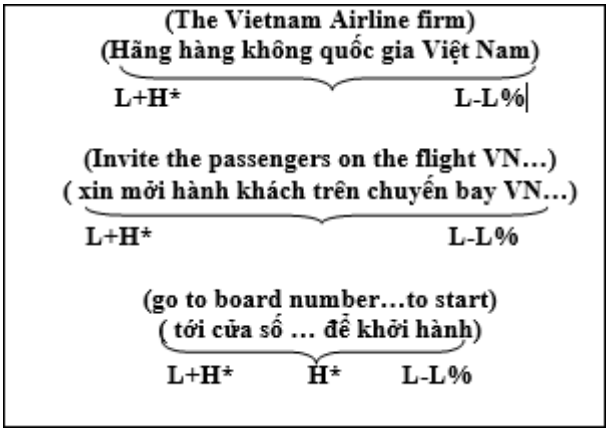
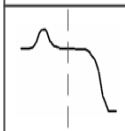
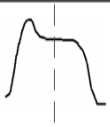
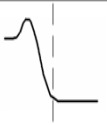
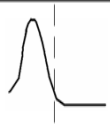
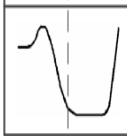
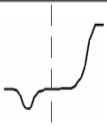
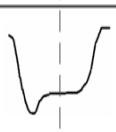


Fig. 2. Pitch accent analysis results of sample airline announcements

Table 2. Relation between intonation and F0 contour

F0 Contour	ToBI	F0 Contour	ToBI	F0 Contour	ToBI
	H* H-L%		L+H* H-L%		L* H-H%
	H* L-L%		L+H* L-L%		H+L* H-H%
	H* L-H%		L* L-H%		H+L* L-H%

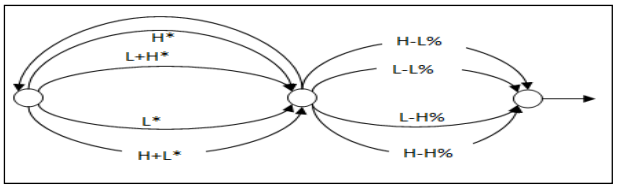


Fig. 3. ToBI grammar in Airline utterance

3 PROSODY IMPROVEMENT IN HMM-BASED SPEECH SYNTHESIS SYSTEM

3.1 *HMM-based Speech Synthesis System*

In HMM-based speech system, speech signals can be reconstructed from feature vectors. A feature vector consists of spectral parameters as Mel-Cepstral Coefficients (MCCs, or Mel-Frequency Cepstral Coefficients-MFCCs), duration, and excitation parameters such as fundamental frequency, F0. To understand the basic notation of HMM-based speech synthesis, we come to four concepts: Spectral modeling, Excitation modeling, State Duration modeling, Language-depent Contextual factors and Context-clustering decision tree.

In Spectral modeling, the MFCCs include energy component and the corresponding delta and delta-delta coefficients are used to represent the spectral. Sequences of Mel-frequency cepstral coefficient vector, which are obtained from speech database using a Mel-cepstral analysis technique, are modeled by continuous density HMMs. It enables the speech to be reconstructed from the coefficients by using the Mel Log Spectral Approximation (MLSA) filter. The MFCC coefficients are obtained through Mel-cepstral analysis by using 40-ms Hamming windows with 8-ms shifts. Output probabilities are multivariate Gaussian distribution [3].

In Excitation modeling, the excitation parameters are composed of logarithmic fundamental frequencies (logF0) and their corresponding delta and delta-delta coefficients. The continuous values in voice region and the discrete values in unvoice region are modeled by Multi-Space probability Distribution. [4]

State duration densities of phonemes are modeled by single Gaussian distributions [5]. Dimension of state duration densities is equal to the number of state of HMM, and the *n*-th dimension of state duration densities is corresponding to the nth state of phoneme HMMs. The duration of each state is determined by HMM-based speech synthesis system. State durations are modeled as multivariate Gaussian distribution [4].

In HMM-based speech synthesis approach, there are many Contextual factors (phone identity factors, stress-related factors, dialect factors, tone factors and intonation) that affect the spectral envelope, pitch and state duration. The only language-dependent requirements within the HTS

framework are contextual labels and questions for context clustering. Some contextual information in Vietnamese language was considered as follows [3]:

Phoneme level:

- Preceding, current and succeeding phonemes.
- Relative position in current syllable (forward and backward)

Syllable level:

- Tone types of preceding, current and succeeding syllables.
- Number of phonemes in preceding, current and succeeding syllables.
- Position in current word (forward and backward).
- Stress-level.
- Distance to {previous and succeeding} stressed syllable.

Word level:

- Part-of-speech of {preceding, current, succeeding} words.
- Number of syllables in {preceding, current and succeeding} words.
- Position in current phrase
- Number of content words in current phrase {before, after} current word.
- Distance to {previous, succeeding} content words
- Interrogative flag for the word.

Phrase level:

- Number of {syllables, words} in {preceding, current, succeeding} phrases.
- Position of current phrase in utterance.

Utterance level:

- Number of {syllables, words, phrases} in the utterance

In many cases, a speech database doesn't have enough contextual samples. In other word, a given contextual label doesn't have its corresponding HMM in the training model set. Therefore, to solve this problem, a Context dependent clustering Decision Tree is applied to classify the phonemes. The question set can be easily extended to include more contextual information which helps the clustering becomes more

detail. The questions are obtained from the phonetic and prosodic characteristics. In training phase, all speech samples of a phoneme with the same context are used to train a 5 state HMM for the context dependent phoneme. In synthesis phase, the decision tree is used to choose an appropriate HMM for each phoneme based on its context.

3.2 *HMM-based Speech Synthesis System*

To improve the naturalness of the synthetic airline statement, we will integrate more context-depend information specified for airline announcement. The adding information is about POS, stress and intonation of the airline utterances.

Figure 4 shows an example of the full context label of phoneme "tr" in the utterance "chúc ngủ ngon" ("good night" in English).

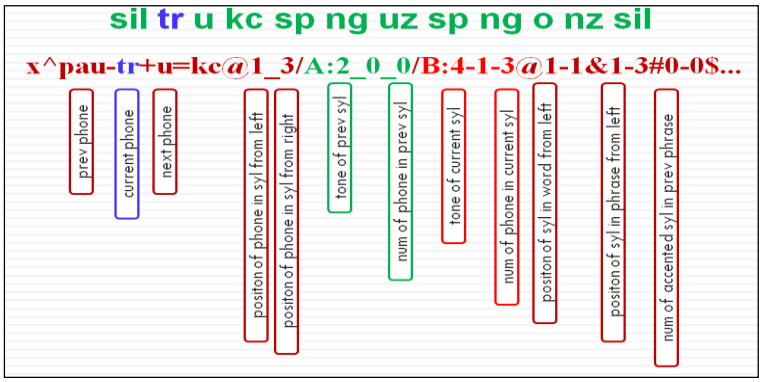


Fig. 4. Full context label of phoneme "*tr*"

Preceding, Current and Succeeding word's POS information is added to the full context label. The information is obtained by using VietTagger and JvnTagger for POS tagging. Figure 5 shows the POS information in full context label.

In intonation transcription, the problem is to identify a melody phrase. The phrases are not always be sentences. Through observing the F0 contour of training speech, the common phrases of the utterances are:

– "*Hãng hàng không quốc gia Việt Nam*" ("The Vietnam Airline firm")

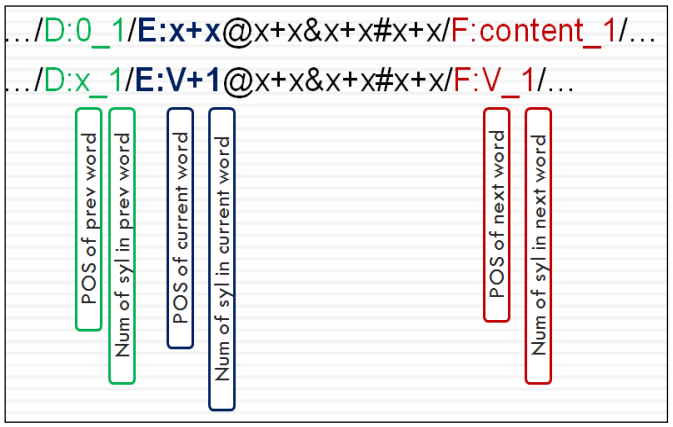


Fig. 5. Added POS information in full context label

- "*xin mời hành khác trên chuyến bay VN___*" ("invite passengers on the flight VN__")
- "*xin mời những hành khách cuối cùng đi__*" ("invite last passengers go to_")
- "*trên chuyến bay__*" ("on the flight__")
- "*tới cửa số__*" ("go to board number__")
- "*trên chuyến bay VN__*" ("on the flight VN_")
- "*khẩn trương tới cửa số__ để khởi hành*" ("hurry go to board number __ to start")
- "*xin cảm ơn*" ("thanh you")

Each of the phrases will have its Pitch Accent and Phrase Intonation Tone. The experiment shows positive result after Intonation transcription is added to full context label. The naturalness is improved.

Stress identification stays a problem in Vietnamese because stress does not play an important role in the language. In Section 2.2, some rules to identify the stress in Airline statement are shown. Based on the rules, the stressed phonemes and information about relative position and number of phonemes are added to the full context label.

4 EXPERIMENTAL RESULTS

In order to evaluate the performance of new system after adding the new Prosodic information, an experiment was established. The experimental setup is shown in Table 4.

Table 4. Experimental setup.

Database	Northern Vietnamese female voice
Training/ test data	510 / 100 sentences
Sampling rate	16 kHz
Analysis window	25-ms width / 5–ms shift
Acoustic features	25 mel-cepstrum, log *F0*, delta and delta-delta
HMM topology	5-state, left-to-right, no skip HMM

4.1 *Subjective Test*

MOS test is used to measure the quality of synthesized speech signals, in comparison with natural ones. The rated levels are: bad (1), poor (2), fair (3), good (4) and excellent (5). In the test, 50 sentences were randomly selected. With 3 types of natural speech, synthetic speech without POS, stress and intonation and synthesized speech with POS, stress and intonation. The number of listeners are 50 people. The speech segments were played in random order in the test. Table 5 shows the MOS test results which were given by all the subjects. The MOS result implied that the quality of natural speech is excellent and the quality of synthetic voice with new prosodic information is better than the synthetic voice without the kind of information.

Table 5. Results of MOS test

Speech	Mean Opinion Score
Natural	5
Without POS, stress, intonation	3.26
With POS, stress, intonation	3.98

4.2 *Objective Test*

To evaluate the synthesis quality, Mel Cepstral Distortion (MCD) [8] is computed on held-out data set. The measure is defined as in Equation 1:

$$MCD=(10/ln10)\sqrt{2*\sum_{i=1}^{25}(mc_i^t-mc_i^e)^2}, \quad (1)$$

where mc_i^t and mc_i^e denote the target and the estimated mel-cepstral, respectively. MCD is calculated over all the MCEP coefficients, including the zeroth coefficient. Lesser the MCD value the better it is. Through observation, it's realized that a difference of 0.2 in MCD value produces difference in the perceptual difference in quality of synthetic speech. Table 6 shows the result of MCD evaluation.

Table 6. MCD results in comparison of 2 synthetic voices with natural voice.

Synthetic voice x Natural voice	MCD
Without POS, stress, intonation x Natural voice	6.47
With POS, stress, intonation x Natural voice	5.86

ACKNOWLEDGEMENTS This work was partially supported by ICT National Project KC.01.03/11-15 "Development of Vietnamese – English and English – Vietnamese Speech Translation on specific domain". Authors would like to thank all staff members of Department of Pattern Recognition and Knowledge Engineering, Institute of Information Technology (IOIT) - Vietnam Academy of Science and Technology (VAST) for their support to complete this work.

REFERENCES

1. Doan, T.T.: Vietnamese Acoustic, Vietnamese National Editions, Second edition (2003)
2. Phan, T.S., Dinh, A.T., Vu, T.T., Luong, C.M.: An Improvement of Prosodic Characteristics in Vietnamese Text to Speech System. In: Knowledge and Systems Engineering, Advances in Intelligent Systems and Computing, vol. 244, Springer, pp. 99–111 (2014)
3. Vu, T.T., Luong, M.C., Nakamura, S.: An HMM-based Vietnamese Speech Synthesis System. In: Proc. Oriental COCOSDA, Urumqi, China, pp. 108–113 (2009)
4. Tokuda, K., Masuko, T., Miyazaki, N, Kobayashi, T.: Multi-space Probability Distribution HMM. In: IEICE, vol. E85-D, no. 3, pp. 455–464 (2002)
5. Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T.: Duration Modeling in HMM-based Speech Synthesis System. In: Proc. ICSLP, vol. 2, Sydney, Australia, pp. 29–32 (1998)

6. Phung, T.N., Phan, T.S., Vu, T.T., Luong, M.C., Akagi, M.: Improving Naturalness of HMM-Based TTS Trained with Limited Data by Temporal Decomposition. In: IEICE Transactions on Information and Systems, vol. E96-D, no. 11, pp. 2417–2426 (2013)
7. Lam, Q.T., Dang, H.P., Anh, T.D.: Context-aware and Voice Interactive Search. In: Proc. 5th International Conference on Soft Computing and Pattern Recognition (2013)
8. Toda, T., Black, A.W., Tokuda, K.: Mapping from Articulatory Movements to Vocal Tract Spectrum with Gaussian Mixture Model for Articulatory Speech Synthesis. In: Proc. 5th ISCA Speech Synthesis Workshop, pp. 31–36 (2004)
9. Vu, K.B., Trieu, T.T.H., Bui, D.B.: Vietnamese Phonemic System: The Construction and Application. In Vietnamese. In: Proc. Celebration of 25 year establishing the Institute of Information Technology, Vietnam Academy of Science and Technology Conference, pp. 525–533 (2001)

LAM-QUAN TRAN
VNA: VIETNAM AVIATION INSTITUTE – VIETNAM AIRLINES,
121 NGUYEN SON, LONG BIEN, HANOI, VIETNAM
E-MAIL: <QUANTL.VAI@VIETNAMAIRLINES.COM>

ANH-TUAN DINH
INSTITUTE OF INFORMATION TECHNOLOGY,
VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY,
18TH HOANG QUOC VIET STREET, HANOI, VIETNAM
E-MAIL: <TUANAD121@GMAIL.COM>

DANG-HUNG PHAN
INSTITUTE OF INFORMATION TECHNOLOGY,
VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY,
18TH HOANG QUOC VIET STREET, HANOI, VIETNAM
E-MAIL: <PDHUNG3012@GMAIL.COM>

TAT-THANG VU
INSTITUTE OF INFORMATION TECHNOLOGY,
VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY,
18TH HOANG QUOC VIET STREET, HANOI, VIETNAM

International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 165–193
Received 17/03/2014, accepted 21/06/2014, final 14/10/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

Towards a Hybrid Approach to Semantic Analysis of Spontaneous Arabic Speech

CHAHIRA LHIOUI,[1] ANIS ZOUAGHI,[2] AND MOUNIR ZRIGUI[3]

[1] *Gabes University, Tunisia*
[2] *Sousse University, Tunisia*
[3] *Monastir University, Tunisia*

ABSTRACT

The automatic speech understanding aims to extract the useful meaning of the oral utterances. In this paper, we propose a hybrid original method for a robust automatic Arabic speech understanding. The proposed method combines two approaches usually used separately and not considered as complementary. This hybridization has the advantage of being robust while coping with irregularities of oral language such as the non-fixed order of words, self-corrections, repetitions, false departures which are called disfluencies. Through such a combination, we can also overcome structuring sentence complexities in Arabic language itself like the use of conditional, concession, emphatic, negation and elliptical forms. We provide, in this work a detailed description of our approach as well as results compared with several systems using different approaches separately. The observed error rates suggest that our combined approach can stand a comparison with concept spotters on larger application domains. We also present, our corpus, inspired from MEDIA and LUNA project corpora, collected with the Wizard of Oz method. This corpus deals with the touristic Arabic information and hotel reservation. The evaluation results of our hybrid spontaneous speech analysis method are very encouraging. Indeed, the obtained rate of F-Measure is 79.98%.

1 INTRODUCTION

The context of our work is the automatic Spoken Understanding Language (SLU) and finalized Human/Machine Communication (CHM).

Various kinds of linguistic knowledge are needed for the proper functioning of an SLU module. This linguistic knowledge can be of several types: lexical, syntactic, semantic, pragmatic and sometimes prosodic for systems taking as input a transcribed text. Many forms are offered to describe these linguistic knowledge kinds among them we can mention: context-sensitive grammar [16], cases grammar [11], Hidden Markov Models [9], Neural Networks [28], N-gram language models [30], λ-calculus [32], logical [17] or Unification Grammars [1] in his various forms (UCG [19]: Unification Categorial Grammar, APSG [31]: Augmented phrase Structure Grammar, LTAG [22]: Lexicalised Tree Adjoining Grammar, STAG [26]: Semantic Tree Association Grammar).

Moreover, we distinguish essentially in automatic SLU two types of approaches for the treatment of linguistic knowledge which are: linguistic approach [18] and stochastic approach [24]. In both cases, the statement is divided into word groups. These groups are frequently called concepts [9, 24] for stochastic approach and chunks [7] for a rule-based approach.

Whereas, these two approaches frequently used separately, enable a more or less effective understanding when dealing with oral speech. In fact, when a speaker speaks in a spontaneous way, the syntax or grammar errors are much more common in spoken than in written language. In one side, this problem is not guaranteed by a detailed linguistic approach. Besides, linguistic approach, as complete as they are, requires a grateful work to analyze corpus by experts in order to extract concept spotting and their predicates. This method is limited to specific fields using restrictive language. Thus, in the case of relatively opened field, it leads to many difficulties like portability and extension which are guaranteed by stochastic approaches.

In the other side, unexpected oral structures do not also obey to any statistical law and may not be satisfactorily modeled by a stochastic approach [23]. Besides, stochastic models do not seem to be able to solve the problem of a detailed language analysis. Added to that, the current stochastic models based on the only restricted linguistic entities observation sequences (words, syntactic categories, concepts, etc.)

cannot encounter statement deepness structures [23]. Finally, learning or training statistical techniques require a large amount of data annotated for learning stochastic model which is not always available.

However, it would be absurd to reject utterances which are syntactically incorrect because the goal is not to check the conformity of user's utterances to syntactic rules but to extract rather the semantic content. Hence, semantic level is important to control meaning of utterances and both syntax and semantic should have to be controlled in order to do not cause understanding problems. Stochastic models are more permissive than linguistic one based on formal grammars. They accept all the sentences of a language. Even incorrect sentences are accepted.

Qualities and drawbacks of these two separately-used approaches allowed us to detect certain complementarities between both of them. This observation is the motivation of our work that attempts to combine the two mentioned-above approaches to take advantage of their strengths. In this regard, we have proposed a hybrid approach based on linguistic and probabilistic methods to the semantic analyzer development for standard Arabic uttered sentence. This is achieved by integrating linguistic details describing local syntactic and semantic constraints on a Hidden Markov Model (HMM) stochastic model. The later materializes the language model of our Touristic Information and Hotel Reservations (TIHR) application.

2 Related Works and Motivation

Oral utterances are so often ungrammatical or incomplete and therefore an important part of the information contained in their texts is lost during an only syntactic rule-based analysis. That is why the analysis covering only syntax aspects is generally not effective. Added to that, Automatic Speech Recognition (RAP) systems generate a significant number of errors not contoured by grammars. Thus, to deal with all these oral treatment problems, some propose either a detailed linguistic phenomena analysis such as in [3], or a combination of a syntactic and semantic analysis such as in [31]. The others resort to stochastic methods as they are more robust to the transcription errors and adapt better to the specificities of oral language. Among these works, we cite for example the work of [9] which aims at achieving a stochastic conceptual decoding based on HMM with two levels in the context of robust spontaneous

speech understanding. In the same context, [25] used also a Bayesian stochastic approach to speech semantic composition in Human/Machine dialogue systems.

Unlike Latin languages, the automatic understanding of spontaneous Arabic speech using stochastic approaches receives little attention at scientific research. In our knowledge, we note that the only work related to the use of stochastic techniques for the spontaneous Arabic language understanding are those of [35]. They have used a stochastic language model for spontaneous Arabic speech semantic analysis in the context of a restricted field (Train Information).

Thus, the originality of our work is to combine two common approaches used to be treated separately (linguistic and stochastic approaches) to the development of a semantic analyzer for standard Arabic utterances. This semantic analyzer is dedicated to tourists who communicate with the Interactive Voice Server (IVS) using Standard Arabic language in order to learn about touristic information that concerns them. Its principal role is then the construction of semantic representations for their utterances.

This hybrid approach consists, in fact, on integrating linguistic constraints in stochastic HMM model. Linguistic analysis should not inhibit the understanding process on the pretext that the input data are grammatically incorrect. According to the Blache citation appeared in [7], shallow parsing is a linguistic technique which facilitates greatly language processing. It rather conforms to spoken irregularities since it is only interested in extracting of word pieces containing only useful information (which are called chunks). Hence, a shallow parsing technique was chosen in our case.

Therefore, results of the linguistic analysis are syntaxico-semantic rules that governing chunks constituting user utterances. These syntaxico-semantic rules play the role of linguistic constraints applied in the first step of our hybrid analysis strategy. The second step will obviously be a stochastic HMM-decoding and then, HMM observations will be these extracted syntaxico-semantic constraints.

The choice to start with a linguistic analysis is appreciated from the fact that the latter does not allow if statements are utterly invalid. Hence, starting with a linguistic one prevents the spread of fatal errors between analyzer modules set in pipeline (see Fig. 1).

We also note that our resulting hybrid model shows a difference from that of Bousquet [8]. This difference consists in having a 3-level HMM

instead of a 2-level Bousquet HMM. These three levels describe respectively three data types: syntaxico-semantic rules, conceptual and probabilistic information. Known that our application domain (TIHR) is relatively open, given its richness in terms of concepts, we numbers 83 between concepts and conceptual segment (see Section 4.3) and 163 linguistic rules (see Section 4.2).

3 Difficulties in Semantic Analysis of Arabic Speech

3.1 *Disfluencies in Spontaneous Utterances*

As indicates their Etymology, disfluencies mean any interruption or disturbance of influence [10]. In what follows, we focus on the major phenomena of disfluencies i.e. self-corrections, repetitions and hesitations.

- Self-correction: this case appears where the speaker made one or more mistakes and corrects in the same statement and in the real time. In this case, the wrong word (or words) is completely pronounced [8].
- Repeating: this is the case of the repetition of a word or series of words.
- Hesitation: it is a break filled in oral production which can be manifested in various ways: by using a specific morpheme (e.g. 'آم' (Eum), 'آه' (Euh) etc.) or in the form of a syllable elongation [10].

3.2 *Semantic Difficulties in Arabic Language Analysis*

Arabic statements semantic analysis is a very difficult task given its semantic richness. This complexity is due to Arabic language specifics, which are:

- Arabic word can mean an entire expression in English or French [7]. For example the word "أرأيت؟" (ara ayta) designs in English language 'Did you see?". Thus, the automatic interpretation of such words requires a prior segmentation which is not an easy task [35].
- Words order in Arabic sentence is relatively variable compared, for example, to English or French languages where words order arrangement in a sentence respects perfectly the sequencing SVO (Subject Verb Object). Whereas, in Arabic, we always have the liberty

to begin with terms in order to put the stress on. In addition, Arabic oral statements do not generally respect any grammar because of the oral spontaneous character. This complicates the task of building a grammar rules to be used in the semantic interpretation process. To do this, we must provide in such grammar all possible combination rules of words order inversions in a sentence.

- The non-capitalization of proper nouns, acronyms and abbreviations makes identification of such words even more difficult than the Latin languages.
- The connection without space of the coordination conjunction 'و' (and) to words. This makes it difficult to distinguish between 'و' as a word letter (e.g. وقف(stand)) and the 'و' having the role of a coordinating conjunction. But this type of combination plays an important role in the interpretation of a statement by identifying its proposals.
- The pronunciation nature of some Arabic letters, for example: غ(gh: ghayn pronunciation). These phonemes have no equivalent in other languages, such as, for example, French or English. Besides, some letters of the Arabic language, such as: ف(f: Fa pronunciation), ح(h: Hha pronunciation) خ(kh: Kha pronunciation) ض(d: Dad pronunciation), ذ(d: pronunciation Thal) ظ(z Zah pronunciation), are pronounced by a strong expiration; so the quality of the microphone can affect speech recognition results;
- The possibility of existence of many graphemes for the same phoneme (e.g. graphemes and ض ظ), or several realizations for the same phonetic grapheme (e.g. grapheme 'ل' (lam) has two different output sounds, depending on letters that precede and follow, as in: 'بالله' (please) and 'الله' (god)). Some graphemes may not be considered in pronunciation ('ـا' for the elongation sound). This phenomenon makes difficult a speech recognition task and, consequently, the task of understanding.

4 Our Hybrid Method

To carry out a robust Arabic statements semantic analysis task, we decided to go through the two following steps:

- Chunking. This step is done by using a shallow rule-based parsing. Such analysis can often make a partial parsing in order to extract only

essential elements that construct statements. These elements are called the chunks [13]. This analysis is guided in our case by a syntactic-semantic Probabilistic Context-Free Grammar (PCFG). The choice of opting for a PCFG is dictated by its adaptability to oral grammatical irregularities. In fact, in ASP (Automatic Speech Processing) field, the PCFG is often used in oral treatment cases. The result of chunking step is set of local linguistic constraints which govern extracted chunks.

– Stochastic analysis. During this step, a stochastic semantic decoding module transforms linguistic constraints extracted in the previous step on concepts.

The following illustrative diagram describes our hybrid semantic analysis strategy adopted in Arabic statements semantic analysis.

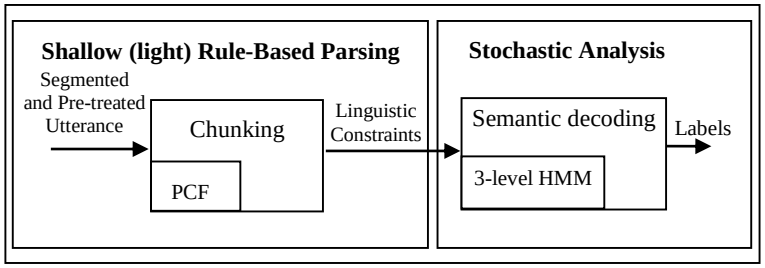


Fig. 1. The hybrid semantic analysis process

In the sequel, we detail each step separately.

4.1 *Segmentation and Pretreatments*

Because of the spontaneous utterance nature that contains various types of disfluencies (see Section 3), an oral statement is inherently rigid and difficult to control. These disfluencies are frequent phenomena that appear normally in spontaneous speech. Here is an example of hesitation and self-correction statement [12]:

هل يوجد مطعم خاص بالكباب هنا آه عفوا بالبيتزا

Is there a restaurant special kabab here, ah pizza sorry?

All these phenomena lead to the ambiguity problems. That is why a pretreatment step is required. This later removes duplication and

unnecessary information, to divide complex utterances into elementary ones, to convert numbers written in all letters, and to determine the canonical forms of words [4, 14].

4.2 *Chunking or Shallow Rule-based Parsing*

The basic idea of this analysis is to limit the depth and richness of a full parsing for the essentials. It consists on a partial parsing that promises to get minimal linguistic structures. These structures are called chunks. They are useful such as in other applications in that they promote their portability.

Our choice to use the chunk notion is based on the Blache citation which asserts that "chunks facilitate the analysis of a statement" [7]. To achieve this, we use the following chunk definition:

Definition1. A chunk is defined as a result of non-recursive categories formed of a head, to which function words and adjacent modifiers may be added [6].

According to the definition 1, we consider a chunk as typical segment consisting of a full word surrounded by functional words and adjacent modifiers. These are connected to the full word by the following linguistic rule:

$$\text{Chunk\_Name} \rightarrow \text{NP}$$
$$\text{NP} \rightarrow \text{Det (Num) (Adj) Name}$$

where the full word appears as a Name, the functional word is the Det and adjacent modifiers are (Num) and (Adj).

Shallow rule-based parsers are often characterized by a two-step process: pattern recognition step and word sense disambiguation one. Similarly, we envisage two independent steps for the realization of our shallow parser.

- Chunk identification
- Chunk attachment (through linguistic rules)

Chunk Identification. The main problem in a shallow rule-based parsing is the recognition of chunks. To resolve it, we applied a pattern recognition algorithm used by many robust parsers [13] from which they take their robustness. The main idea of our algorithm is to loop the input

string n times while calculating at each time the activation chunk degree present in this statement. Given that, according to what has been described in the cognitive ACT-R theory, it is possible to characterize a chunk by its activation level. This activation is described by the following equation extracted from [7]:

$$A_{ik} = B_i + \sum_j w_j S_{ji} \quad (1)$$

where $k \leq n$. In this formula, B represents the latency degree since the last access to the chunk. It is known as the basic activation and stores the frequency and the history of this chunk access. W corresponds, to the items or terms weight which are associated to the chunk. These weights are known as the 'sources' that can activate considered chunks.

Chunk Attachment. Relationships activating a chunk can be viewed as attachments properties that we seek to maximize. Solving chunk attachment problem is reduced to the study of two attachment types: intra-chunk attachment, which concerns words order inside the chunk, and inter-chunk attachments that defines relationships between different chunks constituting statements.

Intra-Chunk Attachment. The first attachment form affects words order inside a chunk. The example below shows a syntactic ambiguity of prepositional group attachment [with mayonnaise PP] either to the verbal group [want to eat VP] or to the nominal group [kebab NP]. This syntactic ambiguity causes a semantic ambiguity since we know more then, if the kebab is mixed with mayonnaise or the man asks two separated entities; one is the kebab and the other is the mayonnaise.

أريد أكل الكباب بالمايونيز
I want to eat kebab with mayonnaise.
(the kebab and mayonnaise are baked together) or
(the kebab and the mayonnaise are baked separately)

Lexical ambiguities are compounded by the attachment ambiguities, especially syntactic ambiguities. In the dialog shown in Fig. 2, the initial user statement causes the system problem because, according to its knowledge, the lexical item *Richard* can be categorized either as a noun or as a surname. The rest of the utterance is analyzed without difficulty.

System:	This is VIS system, hello, what is your request?
User:	It is Richard, I would like to have some information about the X hotel reservation.
System:	Is Richard a name?
User:	Yes, it is.
System:	Ok, what do you want exactly?

Fig. 2. A lexical ambiguity resolution

Regarding the resolution of lexical ambiguities, it can be assigned to the system in order to create an action. This action gives a communicative purpose that leads to the statement “Is 'Richard' a name?”. The user confirmation of this hypothesis allows the system to finish the analysis of the initial user statement, and the search in the user's message box. Then we can say here that the interaction is closed by the user.

However, semantic ambiguities can occur even as in expressions where there are no syntactic or lexical ambiguities. For example, "the coast road" can be the road that follows the coast or the road that leads to it.

One example of syntaxico-semantic intra-chunk ambiguities are uses of condition constraints when the condition chunk ‘لا .... إلا’ (<not> ... <only>) is scattered on two fragments: الشرط and جواب الشرط where ' الشرط' (the proposal) is the portion that lies between locution fragments <not> ... <only> and ' جواب الشرط ' (the proposal response) is the portion that comes after <only>. The example below illustrates what we are trying to explain:

–لا أريد السفر إلا على متن طائرة

I do not want to travel only by a plane

where the use of the negation form in the proposal leads to a positive form in the proposal response to emphasize it. This complex formulation can easily and successfully be resolved by linguistic rules.

Inter-Chunk Attachment. The second form of chunks attachment concerns, in this case, the rearrangement of chunks within the same proposal. This problem increases with the Arabic language where words order is generally not fixed (see Section 3.2). In fact, the sentence in the Arabic language do not undergo the SVO (Subject, Verb, Object) form, as it is the case of French or English languages. We cite two main

categories which are the basis for the construction of Arabic[1] sentences: the first category is SVO (Subject Verb Object) that concerns nominal sentences, while the second, VSO (Verb Subject Object), is a form that describes verbal phrases. This is particularly critical in the case of Arabic language since it is morphologically rich.

– سوسة مدينة ساحلية(جملة إسمية)
Sousse is a coastal city (Nominal sentence)
– أريد السفر إلى سوسة(جملة فعلية)
I want to go to Sousse (Verbal sentence)

Several tricks can also inhibit an automatic understanding of this language. Among them we can mention: reversals (تقديم وتأخير) of sentence constituents as it is the case of nominal sentences where the proposal (الخبر), putted as a prepositional phrase, may precede the theme (المبتدأ). The example below illustrates this reversal that results after the preposition placement of (حرف الجر) 'إلى' putted at the beginning of the subject:

– أريد السفر إلى X
I want to travel to X
– إلى Xأريد السفر
To X I want to travel

A Constraints Grammar. A solution adapted to the intra and inter-chunk attachment problem is the description of different attachment constraints by a set of linguistic rules. This set is called "selection constraints". In this regard, we have designed and specified a probabilistic constraints grammar (see table below) with a probability distribution on the set of potentially finite rules defined by:

$$p(t) \geq 0,$$

$$\sum_{c \in T_G} p(c) = 1,$$

where t denotes the derivation tree which generates the constraint selection $c \in V^*$ and T_G designs all the derivation trees with G is a grammar describing all these constraints.

[1] The problem is the same for other source languages such as Chinese or Russian, and the solutions proposed here will therefore apply in other contexts.

Table 1. Definition of probabilistic constraints grammar

Sets	Description	Statistics
N: Set of non-terminal symbols	Syntactic, semantic and lexical terms : SN, SV, ADJ, VDestination, etc.	91
V: Set of terminal symbols	Linguistic constraints: e.g. C_El_Jar, cause/consequence, condition, WaElMaîya constraint.	43
S: Axiom	Chunk_attach	–
R: Set of production rules	The set of rules r_i linking the constraints	163
p_i: Probability attributed to the r_i rule	A positive coefficient attributed to each rule r_i	–

The elaboration of grammar constraints consists in the definition of:

- The inter-chunk syntax: this is equivalent to indicate how chunks can be arranged together. In our case, this information is modeled by bi-gram transitions that relies different chunks on each other.
- The intra-chunk syntax: this concerns words ordering within each chunk. Each chunk is also modeled by an HMM with bi-grams transition probabilities which relies all words that form the chunk.

For example, the linguistic selection constraint characterizing the condition phenomenon and connecting two chunks forming the condition rule is:

Chunk_condition → not C1 only C2
C1 → الشرط
C2 → جواب الشرط

Thus, the probabilistic specified constraints grammar can be represented as a 2-level HMM as described in Figure 3.

4.3 *Stochastic Analysis*

Our stochastic analysis is based on concepts and conceptual segments notions. Their definitions are inspired from [9]. To achieve this, we used these two following definitions:

Definition 2. A concept (C) is defined as "a general and abstract mental representation of an object", and is independent of the language [8].

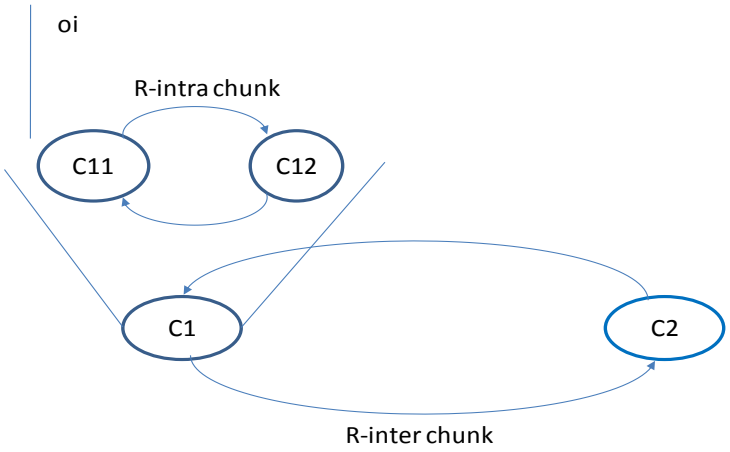


Fig. 3. The representation of constraint grammar on a 2-level HMM

Definition 3. A conceptual segment (SC) comprises word sequences expressing the same meaning unit.

A word sequence making a conceptual segment is an instance of this conceptual segment. For example, the two following word sequences "at five o'clock" and "about four or five o'clock" are two instances of the Time conceptual segment.

According to Definitions 2 and 3, we conclude as a Corollary 1 that a concept can be instantiated by several conceptual segments. In addition, according to Definitions 1 and 3, we conclude as a Corollary 2 that a conceptual segment can be formed by one or more disjoint chunks.

Example:

أريد[الذهاب /إلى سوسة]

I want [to travel \ to Sousse]

الذهاب (to travel) and إلى سوسة (to Sousse) chunks form the conceptual Travel segment.

Statement Semantic Representation Principles. The user statement E is composed of n sequence words noted m_i:

$$E = m_1\, m_2 \ldots m_n$$

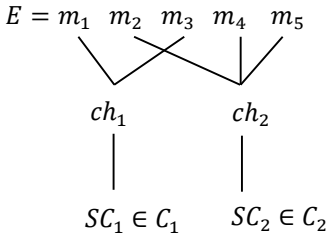
Then, we suppose the following hypothesis: Any statement E expresses a suite of k concepts. These concepts are also expressed using a suite of l conceptual segments and each conceptual segment is formed by a sequence of j chunks.

$$C = C_1\, C_2 \ldots C_n$$
$$SC = SC_1\, SC_2 \ldots SC_l$$
$$SC_i = ch_1\, ch_2 \ldots ch_j$$

The statement E is linearly decomposed into series of l conceptual segments, and hence of $j \times l$ suites of chunks.

$$E = SC_1\, SC_2 \ldots SC_l = (ch_j) \ldots (ch_j)_l$$

For example, if the statement E has five words denoted m_i ($1 \leq i \leq 5$), and each m_i belongs to a chunk ch_k ($1 \leq k \leq 2$), and if E is divided into two conceptual segments SC_1 and SC_2 where each one belongs to a proper concept, a possible representation of the statement E is:



The principle of the statement interpretation with this representation is reduced then to divide into chunks, SC and concepts. This modeling with three levels of knowledge (chunks, SC and concepts) can be represented using a 3-level HMM.

Determination of Concepts and Conceptual Segments. The creation of a model language of a given application requires the determination of all concepts and SC that exist in this application domain. Detailed analysis of the corpus allows determining how these concepts are expressed in his different statements and what their corresponding SC refers to.

However the relationship between concepts and SC is not always so obvious. In fact, many possibilities are offered to the application designer. Added to that, SC can overlap and belong to multiple concepts. We postulate that the meaning of a word depends on the SC where it is located. For example, consider the following two statements:

مدينة الانطلاق سوسة

"Departing from Sousse" (A)

في اتجاه سوسة

"On the way to Sousse" (B)

The word سوسة (Sousse) exists in both statements but should be interpreted as a departure city in the statement (A) and as a destination city in the statement (B). So, according to the conceptual modeling principles of these statements, سوسة (Sousse) belongs to Departure conceptual segment in the statement (A) and to a Destination conceptual segment in the statement (B).

Another example that shows that the meaning of a word depends on the SC to which it is located. Consider the following statement:

العاشرة و عشرة دقائق من فضلك

"A ten and ten minutes please" (C)

The two numbers, in the statement (C), belong to the same concept Time and yet they must be interpreted differently. Indeed, the first number belongs to the Hour conceptual segment and the second belongs to Minute one.

Similarly, the interpretation of SC that compose the statement may depend on one other. Consider the following statement:

لا أريد سوسة مدينة الانطلاق

الرفض الانطلاق

"I do not want Sousse as a starting city" (D)

Refuse Departure

The interpretation of chunks belonging to the Departure conceptual segment depends on chunks presence in the first SC.

Principles that we select for word interpretations in the context of the statement are:

- Word interpretation depends on the chunk where it is located;
- Chunk interpretation depends on the SC where it is located;
- SC interpretation of a SC depends on the concept from which it is issued;
- Word interpretation can be modified by the presence of previous or next chunks in the statement.

Semantic Analysis Principal. Semantic analysis principles are useful to interpret every word that composes statements according to chunks and SC to which they were assigned. The interpretation may eventually be modified later by the presence of previous or subsequent chunks.

The two tables below illustrate, in terms of concepts, SC and chunks, the conceptual domain of our application that deals with Tourist Information and Hotel Reservations (TIHR) (see Section 5.1).

Table 2. Statistics of concepts, SC and linguistic constraints in our 3-level HMM

Concept (1-Level HMM)	Conceptual Segment (2-Level HMM)	Constraint rule (3-Level HMM)
3	80	163

Table 3. Identification of concepts, several SC and their corresponding chunks for the TIHR field

Concepts	Conceptual Segments (SC)	Chunks
Opening Closing Dialog	Request	{« أريد » (I want), « أطلب » (I demand), « أرغب » (I desire), …}
	Accept	{« أقبل » (I accept), …}
	Refuse	{« أرفض » (I do not accept), …}
Hotel Reservation	Hotel	{« نزل » (hotel), « فندق » (hostel), …}
	Room	{« بيت » (room), …}
	Tariff	{« كلفة » (cost), « ثمن » (price), « سعر » (quote), …}
	Number	{« عدد », …}
Touristic Acknowledge-ment	Living_City	{« مدينة » (city), « بلدة » (town), « قرية » (village), …}
	Itinerary	{« طريق » (road), « الثنيا » (route), …}
	Transportation Means	{« وسيلة النقل » (mean of transport), …}
	Price_Ticket	{« ثمن التذكرة », …}

Word interpretation depends on the SC where it is located follows the two principles below:

- Word interpretation depends on the concept that issued it;
- Word interpretation may be modified by the presence of a previous conceptual segment or according to the statement.

Semantic statement interpretation principles are useful to interpret every component in this statement according to word classes and SC to which it was assigned at the conceptual decomposition. The interpretation may possibly be changed later by the presence of previous or subsequent conceptual segments.

The 3-Level HMM Elaboration. The statement semantic analysis requires linguistic skills that are represented by a language model. According to our hybrid approach, our language model is defined in terms of concepts, SC, and chunks attachment rules and hence modeled by a 3-level HMM. The observations of the model will be chunks simulated by linguistic constraints. The latter were extracted by shallow parsing method based on PCFG analysis realized in the former step (see Section 4).

HMM Basic Principle. HMM models are powerful statistical tools that have been successfully used in various fields such as Automatic Speech Recognition (ASR) and Dialog Management (DM). They allow to model observation sequences putted in their two different forms; discreet and continuous form. Thus, we chose to represent our language model using an HMM model.

HMM are also N-grams language models. That is mean that the symbol emission probability depends on N-1 previous symbols. In our case, we consider that our HMM is a bi-gram language model.

The HMM model topology must be designed by a field expert of the considered application. Nevertheless, HMM parameters are automatically learned from training data. Therefore, stochastic models are more portable than linguistic ones where all rules must be explicitly written.

A 1-level HMM consists on two processes: the first is observable and the second is hidden. In all cases, following hidden states forms a Markov chain of order 1. A multi-level HMM is a model where

observations of each hidden state are also modeled by a Markov model. The embedded Markov models number defines the model level number.

Retained Modeling. To be able to represent all the information that our language model carries out (linguistic rules, concepts and SC), a 1-level HMM is insufficient. We propose then a 3-level model (see Figure below) where: the first level is described by an HMM whose states represent our application concepts. According to the Table2. we have in total 3 principal concepts so, they correspond to three 1-level hidden states. Each concept is represented, in its turns, by an HMM whose states correspond to the conceptual segments (SC_i see Table 2.). Each SC will be represented by an HMM describing linguistic constraints of its realization.

Fig 4 shows an overview of our 3-level designed HMM.

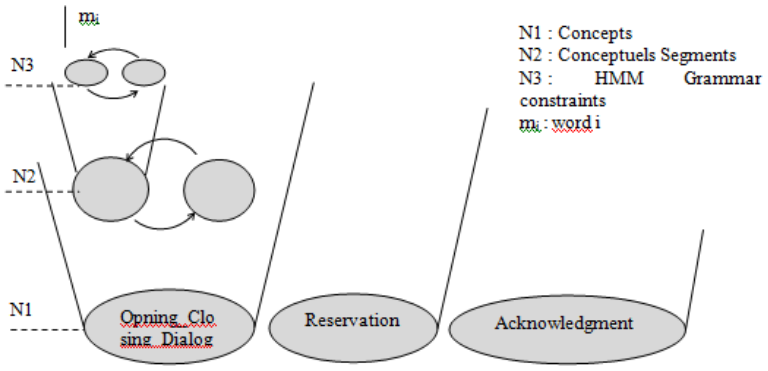


Fig. 4. Overview of our 3-level HMM

5 THE FINALIZED CONSIDERED APPLICATION

To test our application and estimate the HMM parameters, we use a corpus dedicated to the study of touristic applications accessing to databases. The main reason for choosing this application field is the statistically representative size of the training corpus that we have. Moreover, through this corpus, we have the opportunity to produce an Arabic dialogue corpus in the same manner of those which are produced within the Francophone and Anglo-Saxon projects such as MEDIA [15]

and LUNA [24]. Table 4 below shows a comparison between our Arabic dialog corpus and some other oral corpora issued in other fields and in other languages.

Table 4. Comparison between several International corpora and the our TIHR Corpus

Corpus	Language	Field	Size (on statement)
MEDIA	French	Touristic Reservation	18K
LUNA	Polonais	Transport Information	12K
TELDIR	Allemand	Time Train	22K
ATIS	English	Plan Ticket Reservation	6K
PlanRest	French	Restaurant Reservation	12K
TIHR Corpus	Arabic	TIHR	35K

5.1 *Corpus Establishment*

Oral corpora represent a significant proportion in the development of the automatic Spoken Language Understanding (SLU). These are closely linked to the availability of such corpus. However, Arabic speech corpora that deal with 'Touristic Information and Hotel Reservation' are very rare or even unavailable. Thus, to carry out our experiments in the context of this work, we were to build our own study corpus.

Our corpus is derived from the simulation of tourist information server and hotel reservations. Dialogues enunciated by tourists are in Standard Arabic Modern language (ASM). In fact, we consider that tourists are people who do not have Arabic as a native language. That is why their conversations were be uttered in the ASM language and not in common Arabic parlance.

These dialogues are about different themes such as: choice of living city, finding a route or a tourist event, a satisfaction of a price or date constraint. They were held between humans and machines through the Wizard of Oz protocol (Wizard of Oz, WoZ). Indeed, during the exchange, users believe converse with a machine while the dialogue is actually supported by a human operator that simulates information and reservation server responses. The operator is assisted by the WOZ tool in the generation of responses to provide the user. After each user sentences, the operator shall consult the WOZ tool that offers to provide the answer to reflect the new dialogue state. To diversify the operator's responses, the WOZ tool is set at the messages, instructions and

scenarios. A message sets are associated with the application to vary response formulations. With each call, the operator must follow instruction series (e.g. pretending not to have understood users to simulate mistakes which would make a real system). These instructions must be provided to the WOZ tool and depend on the scenario chosen for the dialogue recording [12].

A detailed description of technical corpus characteristics is given in Table below.

Table 5. Characteristics of our corpus (TIHR Corpus)

	Dialog Number	Speaker Number	Average Utterance Number /Dialog	Word Total Number
TIHR Corpus	10 000	1000	10	730 000

All queries corpus was recorded and transcribed manually as transcription standards in XML files, and labeled according to standards proposed by the ARPA community. Manual transcription was made a loyal way that it was recorded. That is to say, words are transcribed as we hear in recordings. Hence, we note disfluencies presence such as hesitations, self-corrections, repetitions, fault departures, etc.

In our corpus, we distinguish three query types i.e. independent query context, dependent query context, and absurd requests. The study of the corpus has allowed us to identify three concepts namely, Opening_Closing_Dialog, Hotel_Reservation and Touristic_Acknowledgement (see Table 3). In addition, we have defined SC among that we can mention: Living_City, Hotel, Travel, Tariff_Travel, Time_Travel, Price_Ticket, Period_Travel, Itinerary, Transport_Mean, Departure_City, Arrival_City, etc. Each conceptual segment is associated with a chunks set containing reference words relating to our application field. Words constituting chunks are linked by linguistic rules governing intra- and inter-chunk relations (see Table 1).

5.2 *Tests and Evaluation*

To avoid error propagation of the former linguistic analysis (surface analysis) to the second one (stochastic analysis), manual chunk verifications and their produced rules is performed just after the linguistic analysis to evaluate the shallow rule-based parsing results. An

automation of this verification is suggested as one perspective. For the evaluation of the stochastic analysis, Kohavi cross-validation [28] is performed. In fact, this latter is used to validate an existing prediction or classification model or to find the best model by estimating its precision. It helps with a reduced bias to estimate the efficiency measure of our 3-level HMM. This measure is the average of each embedded HMM. The cross-validation technique is useful when the number of observations is fixed.

Cross-validation methods principle. The cross-validation method principle is to divide the observation set into two separate and independent subsets where the first is called training set (that is the greater one) and the second is called validation or test set. The training set is used to generate the appropriate probabilistic HMM model. Nevertheless, the test set is used to evaluate the trained model according to the evaluation criteria.

Cross-Validation Initial Conditions. To perform cross-validation test, three initial conditions have to be satisfied. Indeed, we have to:

- prepare a sample observation
- have prediction model
- mesure the method performance for two separate sets either by calculating the rate error or by measuring the efficiency, precision, recall, F-score.

Cross-Validation Method Choices. Various cross-validation methods exist. Among them we can mention three principal ones that are respectively called: Holdout (also called Test set), Leave-one-out and k-fold. A detailed description of their advantages and disadvantages of each method is found in [5]. Therefore, we have not applied the Holdout method since it does not give good results because the observation set may be small for it. This badly affects the precision of performance values measured for model evaluation. In addition, the method of Leave-out-has been abandoned in our evaluation task because it is the slowest and the less used method. So we are simply satisfied to use just the K-fold method whose principle is to find the compromise of the two pre-cited ones.

K-Fold Method. Knowing that the parameter k often used in practice is k =5 or k =10 [21], the 5-fold and 10-fold are the two respective training methods which are used in our experiments. In both cases, we have divided the observation set in k disjoint partitions of equal size where $k-1$ partitions are used for training and the left partition is used for the test. This process is repeated such as each partition is used one time for the evaluation. We then obtain k models and therefore, k accuracy measures. The final estimated accuracy value of the model learning from the training set is the average value of k calculated accuracies.

The Model Learning Step. In this step, the Baum-Welch learning algorithm uses the training observations and an initial model for generating a decoding model. Learning is to adjust the initial model parameters. The library used in our experiments is Jahmm (JAva HMM implementation) [37]. The latter allows the HMM implementation in Java and contains the basic algorithms for using HMM. To estimate the established HMM efficiency, we used the cross-validation method. This latter allows estimating the classification model effectiveness in general and for HMM models in particular.

Initial Hypothesis. Initial parameters choice for learning or training an HMM presents some difficulties [27]. However, in our case, we treat only discrete symbols. Thus, any observation probabilities distribution is applicable to our model.

Knowing that we have a 3-level HMM, where each level L_i $(1 \leq i \leq 3)$ is characterized by n_i hidden states, we considered $1 + n_1 + n_2$ initial HMM_{ij} $(1 \leq j \leq n_i)$ which designate the ith level and the jth hidden state HMM. The characteristics of several models are shown in the following table. These models are classified as follow:

– HMM_1 for $(n_1 = 3)$ concepts in the first level of the 3-level global HMM;
– $HMM_{21}, HMM_{22}, HMM_{23}$ for $(n_2 = 80)$ conceptual segments in the second level. In this level we have $(n_1 = 3)$ embedded HMM, where n_1 is the state number of the first global HMM level;
– $(n_2 = 80)$ HMMs for $n_3 = 163$ linguistic constraints in the third level. This number is equal to the SC number that the second level contains.

Table 6. Initial models probability distribution features (for the first and the second global HMM level)

	HMM_1	HMM_{21}	HMM_{22}	HMM_{23}
$\boldsymbol{\pi_i}$	Non-uniform	Uniform	Randomly	Non-uniform
$\boldsymbol{A_{ij}}$	Non-uniform	Non-uniform	Randomly	Uniform
$\boldsymbol{B_i(o_t)}$	Uniform	Uniform	Randomly	Non-uniform

The Learned Models. After the learning algorithm execution, the initial models parameters are gradually refined into a finite iteration number. We empirically chose 10 iterations. The latter value corresponds to learning algorithm stopping criterion. Following tables show average values of HMM initial probabilities simulated as two cases of k-fold cross-validation method applied for the 1-level HMM.

Table 7. Initial probabilities average value of the first level HMM with 5-fold cross-validation method

	π_1	π_2	π_3	π_4	π_5
HMM_1	0.2	0.3	0.3	0.15	0.05

Table 8. Initial probabilities average value of the first level HMM with 10-fold cross-validation method

	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}
HMM_1	0.1	0.03	0.09	0.07	0.17	0.11	0.25	0.08	0.02	0.08

5.3 *Evaluation*

In order to calculate the performance criteria and thus to assess the decoded utterance quality, we generate an n-classes confusion-matrix (Table 3). It is obtained by comparing predicted labels sequences (resulting states of the decoding step) and real labels sequence (extracted manually). Real labels sequence is determined by confusion-matrix rows information. As for predicted labels, they correspond to the confusion-matrix columns information (see Table 9).

Diagonal confusion-matrix C_i^j cells represent correct predicted labels. They correspond to the C_i label occurrence number in two state sequences. Besides, confusion-matrix C_i^j cells represent incorrect

predicted labels with $i \neq j$. They correspond to real and predicted C_i and C_j label occurrence numbers.

Result Interpretation. To evaluate the learning generated models, we have produced the most likely predicted states sequence (generated by the learned HMM) associated to a given observation sequence (sequence of real states). This is done for all test set observation sequences.

Subsequently, we built confusion matrices from predicted and real state sequences according to the chosen cross-validation method. Specifically, we have obtained five confusion-matrices using the K = 5 method and ten confusion-matrices using K = 10.

To illustrate this, we present a confusion-matrix for three labels representing the 1-level HMM states. This matrix has been obtained with the method K = 5 on its first test set.

Found: L1 L3 L1 L2 L1 L3 L3 L2 L3 L3

Predicted: L1 L3 L1 L1 L1 L3 L3 L3 L3 L3

Table 9. Example of a confusion matrix to 3 classes

		Predicted		
		L1	L2	L3
Found	L1	3	0	0
	L2	1	0	1
	L3	0	0	5

In the above-shown confusion-matrix, all label L1 predictions are correct. They are equal to three. For label L2, there is no correct prediction. On the other side, for label L3, five predicted labels are correct. We have in total eight correct labels among ten ones. We conclude then that the model generates a satisfying correct prediction numbers. Therefore, the decoding quality is satisfactory.

Performance Criteria. The table below presents precision, recall, and F-score average values for HMM_1, HMM_{21}, HMM_{22} and HMM_{23} embedded trained models of our 3-level HMM using K5, K10 methods. These metrics were calculated from confusion matrices for each test set.

Table 10. Precision, Recall and F-score average values for some embedded trained models using K5, K10 methods.

Modèles	Précision	Rappel	F-score
HMM_1	70.00%	71.00 %	73.79%
HMM_{21}	71.08%	68.99%	74.10%
HMM_{22}	69.98%	70.89%	72.90%
HMM_{23}	70.02%	72.00%	73.77%

Results Comparison. Comparison results between the Bousquet's CACAO stochastic conceptual decoder [9] and the Zouaghi and Zrigui's semantic decoder [36] shows that the error response rate, obtained by our system, was reached 20.02% which is considerably less than Bousquet system (see Table 11).

Table 11. Comparison of our hybrid system results with several well-known systems

	CACAO	ORÉODULE PROJECT DECODER	OUR ANALYZER
APPROACH TYPE	STOCHASTIC	SEMANTIC	HYBRID
% ERROR RATE	29.11%	29%	20.02%

6 CONCLUSION AND PERSPECTIVES

One of most supervised objectives that we hope to achieve when we implemented our semantic analyzer system was to fulfill spontaneous spoken Arabic language robust parsing while making the application field wider than it is currently done. Linguistic approaches are not usually viewed as efficient tools for pragmatic applications. That's why we were interesting in combining two frequently separately-used approaches (linguistic and stochastic approaches).

A second objective was to have rather a generic system, despite a field-based linguistic knowledge use. This constraint is achieved through generic rule definitions as well as their probabilities for the third linguistic integrated HMM level training. This makes it possible to estimate efficiently its parameters. Our analyzer performances show that the two divergent approaches combination can bear comparison with systems which are based on a lonely approach (stochastic approach for

CACAO Bousquet's conceptual decoder and semantic approach for Zouaghi and Zrigui semantic decoder).

The model was trained using the Baum-Welch algorithm. This adjusts the initial model parameters. The learning step was carried out according to different cross-validation techniques. Then, the model evaluation was also carried out in the quantitatively and well-known methods using different metrics such as precision, recall and F-score.

As a perspective, we hope to refine our system analysis to be tested in a second challenge by evaluation campaign that will focus on the phenomena described in the typology that we have proposed.

REFERENCES

1. Abeillé, A.: Les Nouvelles Syntaxes: grammaires d'unification et analyse du français (Linguistique). Paris: Armand Colin, 326 pp. (1993)
2. Antoine, J.-Y., Goulian, J., Villaneau, J.: Quand le TAL robuste s'attaque au TAL parlé : analyse incrémentale pour la compréhension de la parole spontané, TALN 2003, Batz-sur-Mer (2003)
3. Antoine, J.-Y.: Pour une Ingénierie des Langues plus Linguistique", HDR Computer Science, University of South Bretagne, Vannes, France (2003)
4. Bahou, Y., Belguith, H.L., Ben Hamadou, A.: Towards a Human-Machine Spoken Dialogue in Arabic. In: 6th Language Resources and Evaluation Conference (LREC 2008). Workshop HLT Within the Arabic World. Arabic Language and Local Languages Processing Status Updates and Prospects, Marrakech, Morocco (2008)
5. Besbes, G.: Modélisation De Dialogues A L'aide D'un Modèle Markovien Caché Mémoire Présenté, Mémoire présenté à la Faculté des études supérieures de l'Université Laval (2010)
6. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python, O'Reilly Media (2009)
7. Blache, P.: Chunks et activation, un modèle de facilitation du traitement linguistique, TALN-RÉCITAL, Les Sables d'Olonne (2013)
8. Bouraoui, J.-L.: Traitement automatique de dysfluences dans un corpus linguistiquement constraint, Actes de la 16ème Conférence sur le Traitement Automatique des Langues Naturelles, TALN'09, Senlis, France (2009)
9. Bousquet-Vernhettes, C.: Compréhension robuste de la parole spontanée – Décodage conceptuel stochastique. PhD thesis, Université Paul Sabatier (2002)
10. Bove, R.: A Tagged Corpus-Based Study for Repeats and Self Repairs Detection in French Transcribed Speech. Proceedings of the 11th

International Conference on Text, Speech and Dialogue, TSD'08, Brno, Czech Republic (2008)
11. Bruce, B.: Cases Systems for Natural Languages", Artificial Intelligence, Volume 6, pp. 327–360 (1975)
12. Lhioui, C., Zouaghi, A., Zrigui, M.: A combined A Combined Method Based on Stochastic and Linguistic Paradigm for the Understanding of Arabic Spontaneous Utterances. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2013, Part II. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 7817, Springer, pp. 549–558 (2013)
13. Trouilleux, F.: Un analyseur de surface non déterministe pour le français, TALN 2009, Senlis (2009)
14. Hadrich Belguith, L., Bahou, Y., Ben Hamadou, A.: Une méthode guidée par la sémantique pour la compréhension automatique des énoncés oraux arabes. International Journal of Information Sciences for Decision Making (2009)
15. Bonneau Maynnard, H., Mctait, K., Mostefa, D., Devillers, L., Rosset, S., Paroubek, P., Bousquet, C., Choukri, K., Goulian, J., Antoine, J.-Y., Béchet, F., Bontron, O., Charnay, L., Romary, L., Vergnes, N., Vigourous, N.: Constitution d'un corpus de dialogue oral pour l'évaluation automatique de la compréhension hors et en contexte du dialogue, Actes des Journées d'Etude sur la Parole (JEP 2004), Fès, Maroc (2004)
16. Autebert, J.M., Berstel, J., Boasson, L.: Context-free languages and pushdown automata. In: Handbook of Formal Languages, vol. 1, Word, Language, Grammar, Springer (1997)
17. Villaneau, J., Ridoux, O., Antoine, J.-Y.: Comprehension de l'oral spontané Présentation et évaluation des bases formelles de LOGUS RSTI RIA. volume 18, no. 5–6, pp. 709–742 (2004)
18. Allen, J.: Natural language understanding. Redwood City, CA, USA, Benjamin-Cummings Publishing Co., Inc. 28 (1988)
19. Kray-Baschung, K., Bes, G.G., Guillotin, T.: French Unification Categorial Grammars (2014)
20. Kurdi, M-Z.: Contribution à l'analyse du langage oral spontané. Thesis, Université Joseph Fourier, Grenoble (2003)
21. Liu, B.: Web Data Mining – Exploring Hyperlinks, Contents and Usage Data, Springer (2006)
22. Lopez P., Analyse guidée par la connexité de TAG lexicalisées, TALN'98, Paris (1998)
23. Cori, M.: Des Méthodes De Traitement Automatique Aux Linguistiques Fondées Sur Les Corpus, Langages, vol. 171, pp. 95-110 (2008)
24. Meurs, M.J.: Approche stochastique bayésienne de la composition sémantique pour les modules de compréhension automatique de la parole dans les systèmes de dialogue homme-machine. Thesis, Université d'Avignon et des Pays de Vaucluse (2009)

25. Meurs, M., Lefèvre, F., De Mori R.: Spoken Language Interpretation: On the Use of Dynamic Bayesian Networks for Semantic Composition. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP (2009)
26. Kurdi, M.Z.: A spoken language understanding approach which combines the parsing robustness with the interpretation deepness. International Conference on Artificial Intelligence, IC-AI'01, Las-Vegas, USA (2001)
27. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, vol. 77, no. 2, pp. 257–286 (1989)
28. Kohavir, R.: A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, pp. 1137–1143 (1995)
29. Jamousi, S., Smaili, K., Haton, J.P.: Contribution à la compréhension de la parole par des réseaux neuronaux, Quatrième Rencontres Jeunes Chercheurs en Parole (RJC'01), Mons, Belgium, pp. 70–73 (2001)
30. Knight, S., Gorell, G., Rayner, M., Milward, D., Koeling, R., Lewin, I.: Comparing grammar-based and robust approaches to speech understanding: a case study. In: Proceedings of European conference on speech communication and technology (2001)
31. Seneff, S.: TINA: A Natural Language System for Spoken Language Applications. Computational Linguistic, vol. 18, no. 1, pp. 61–86 (1992)
32. Villaneau, J., Antoine, J.-Y., and Ridoux, O.: Logical approach to natural language understanding in a spoken dialogue system. In Sojka, P., Kopecek, I., and Pala, K., editors, The International Conference on Text, Speech and Dialogue, TSD 2004. Lecture Notes in Computer Science, vol. 3206, pp. 637–644. Springer (2004)
33. Brondsted, T.: The Linguistic Components of the REWARD Dialogue Creation Environment and Run Time System. IEEE 4th Workshop Interactive Voice Technology for Telecommunications Applications (IVTTA'98), Turin, Italy, pp. 71–74 (1998)
34. Zouaghi, A., Zrigui, M., Antoniadis, G.: Compréhension Automatique de la Parole Arabe Spontanée. Traitement Automatique des Langues, vol. 49, no. 1, pp. 141–166 (2008)
35. Zouaghi, A., Zrigui, M., Antoniadis, G., Présentation d'un modèle numérique pour la compréhension de la parole arabe spontanée (2007)
36. Zouaghi, A., Zrigui, M., Considération du contexte pertinent pour améliorer les performances d'un étiqueteur sémantique de la parole arabe spontanée, RJC (2005)
37. Online: http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/

CHAHIRA LHIOUI
ISIM OF MEDENINE, GABES UNIVERSITY,
ROAD DJERBA, 4100 MEDENINE, TUNISIA
E-MAIL: <CHAHIRA_M1983@YAHOO.FR>

ANIS ZOUAGHI
ISSAT OF SOUSSE, SOUSSE UNIVERSITY,
TAFFALA CITY (IBN KHALDOUN), 4003 SOUSSE, TUNISIA
E-MAIL: <ANISS.ZOUAGHI@GMAIL.COM >

MOUNIR ZRIGUI
FSM OF MONASTIR, MONASTIR UNIVERSITY,
AVENUE OF THE ENVIRONNEMENT 5019 MONASTIR,
LATICE LABORATORY, ESSTT TUNIS, TUNISIA
E-MAIL: <MOUNIR.ZRRIGUI@FSM.RNU.TN>

Author Index

Elena Lloret
Bernardo Magnini
Cerstin Mahlow
Suresh Manandhar
Diana Mccarthy
Alexander Mehler
Rada Mihalcea
Evangelos Milios
Dunja Mladenic
Marie-Francine Moens
Masaki Murata
Preslav Nakov
Costanza Navarretta
Roberto Navigli
Vincent Ng
Joakim Nivre
Attila Novák
Kjetil Nørvåg
Kemal Oflazer
Constantin Orasan
Ekaterina Ovchinnikova
Ivandre Paraboni
Saint-Dizier Patrick
Maria Teresa Pazienza
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Octavian Popescu
Marta R. Costa-Jussà
German Rigau
Horacio Rodriguez
Paolo Rosso
Vasile Rus
Kepa Sarasola
Roser Sauri
Hassan Sawaf
Satoshi Sekine
Serge Sharoff
Grigori Sidorov
Kiril Simov
Vivek Kumar Singh
Vaclav Snasel
Thamar Solorio
Efstathios Stamatatos
Carlo Strapparava
Tomek Strzalkowski
Maosong Sun
Stan Szpakowicz
Mike Thelwall
Jörg Tiedemann
Christoph Tillmann
George Tsatsaronis
Dan Tufis
Olga Uryupina
Karin Verspoor
Manuel Vilares Ferro
Aline Villavicencio
Piotr W. Fuglewicz
Savas Yildirim

ADDITIONAL REFEREES FOR THIS VOLUME

Mahmoud Abunasser
Naveed Afzal
Iñaki Alegria
Hanna Bechara
Houda Bouamor
Janez Brank
Chen Chen
Víctor Darriba
Owen Davison
Ismaïl El Maarouf
Mahmoud El-Haj
Milagros Fernández-Gavilanes

Daniel Fernández-González
Corina Forascu
Kata Gabor
Mercedes García Martínez
Diman Ghazi
Rohit Gupta
Francisco Javier Guzman
Kazi Saidul Hasan
Radu Ion
Zahurul Islam
Milos Jakubicek
Antonio Jimeno
Olga Kolesnikova
Mohammed Korayem
Tobias Kuhn
Majid Laali
Yulia Ledeneva
Andy Lücking
Tokunbo Makanju
Raheleh Makki
Akshay Minocha
Abidalrahman Moh'D
Zuzana Neverilova
An Ngoc Vo
Mohamed Outahajala
Michael Piotrowski
Soujanya Poria
Francisco Rangel
Amir Hossein Razavi
Francisco Ribadas-Pena
Alvaro Rodrigo
Armin Sajadi
Paulo Schreiner
Djamé Seddah
Karan Singla
Vit Suchomel
Aniruddha Tammewar
Yasushi Tsubota
Francisco José Valverde Albacete
Kassius Vargas Prestes
Tim Vor der Brück
Tadej Štajner