Natutal Language Processing: Perspective of CIC-IPN

A. Gelbukh

Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Mexico City, Mexico
www.Gelbukh.com

***Abstract*—In this paper I outline part of the results of fifteen years of research of the Natural Language Processing (NLP) Laboratory of CIC-IPN. NLP is a set of technologies that in long run will allow for dialogue with robots in our everyday language or for overcoming the language barrier between people. Our group's work is concentrated on the internal tasks of this technology, such as resolving ambiguities and constructing dictionaries.**

Keywords—natural language processing, information retrieval, sentiment analysis, word sense disambiguation, similarity measures

I. Introduction

Natural Language Processing is a set of technologies aimed to allow computers to meaningfully process texts or speech in everyday human language [1]. Such technologies allow conducting a dialog with a robot as one would with a person, automatically translating texts from one language to another, or searching for information in huge collections of texts.

In this short paper I outline part of the results of the Natural Language Laboratory of CIC-IPN. Due to space limitations I will not present our research on such areas as bilingual text alignment [2] or detection of synonymy [3], but only touch upon those directions of our research that I consider especially important: semantics, representation of text, and applications.

II. Semantics

Semantics plays crucial role in advanced applications of natural language processing (NLP). It maps words, phrases or texts to some representation of meaning, being meaning what matters for the users of those applications.

Words and texts are sequences of letters (or sounds when speech is considered). They have such properties as letters of which they are composed, pronunciation, length, grammatical categories, or their relative order in the text. In contrast, meanings concepts, or ideas, which often correspond to objects or phenomena occurring in the world and in human life. They have such properties as being subordinated to each other (for example, a *dog* is an *animal*) or have other relationships that reflect the properties of the corresponding objects or phenomena (*tail* is a part of a *dog*).

Semantic analysis deals with the corresponding between texts and meanings. Meaningful parts of text, such as words or phrases, can have compositional and non-compositional meaning. The non-compositional meaning is a mapping between words or word combinations and semantic units in such a way that the result is not predictable by analysis of the internal structure of the text unit. For example, the word *dog* can be mapped to a meaning referring to the corresponding animal; a word combination hot dog is to be mapped to a meaning referring to a corresponding food item.

Compositional mapping, in contrast, is built as a set of relationships between elemental units constructed in a non-compositional way, thus forming a semantic network—a complete representation of a sentence or text. For example, *John is eating* a *hot dog* is mapped to a network with the nodes corresponding to the meanings of *John*, *eating*, and *hot dog*, with relationships that indicate that *John* is who eats, and *hot dog* is what is *eaten*. There is a great number of frameworks that specify how specifically to represent such nodes and relationships.

Accordingly, semantic analysis of text has two main goals: mapping elementary text units to their elementary meanings, and determining the relationships between them. In this paper I will shortly review our research in three areas of this process: word sense disambiguation, analysis of collocations and lexical functions, and recognizing textual entailment.

A. Word Sense Disambiguation

Word sense disambiguation (WSD) is perhaps conceptually the simplest case of semantic analysis—which does not imply that it is technically simple or easy [4]. It deals with the distinction between word types, words out of context, or dictionary words—words as they are listed in the dictionary (they correspond to the language in Saussure's terms, rules that govern the use of language)—and word tokens, words in the context, or running words —words as they occur in text (speech in Saussure's terms, an instance of use of the language). Dictionary words can have different senses: for example, *dog* has in the WordNet dictionary the senses of animal, frump, cad, pawl, andiron, etc. However, a running word in the text usually has only one intended meaning, as for example, in *the dog barks*. The task of WSD is, therefore, to automatically choose for a running word in context one of the senses given in a dictionary; more specifically, to assign to each running word the number of the corresponding sense.

One of the complications of the task is the fact that the dictionary definitions, which allow for distinguishing the senses, can be so similar that automatically decide which one applies is difficult [5]. Different dictionaries can present this problem to different grade [6].

We observed that different classes of words require different types of algorithms for disambiguation: while "easy" words are best dealt with simple algorithms, for the "tough" words simple algorithms fail, and much more elaborate algorithms are necessary [7]. Therefore, an important task is meta-classification: deciding what kind of algorithm is to be used for what specific words [8].

In spite of being one of the first algorithms suggested for WSD, the Lesk algorithm [9] is an excellent basis for more elaborated algorithm and an excellent model for understanding the effects of various factors. This algorithm consists in three parts. First, a similarity, or relatedness, measure between the definitions of the senses of different words is to be chosen: say, how much the definition of the sense of *dog* as animal is related to the sense of *barking* as animal sound. Next, for each running word in the text to be disambiguated, for each its possible sense, its relatedness to each sense of each other word in the text is defined, using both the relatedness measure between senses and other factors, such as distance. Finally, one sense per each word is chosen so that the total relatedness between the chosen senses be maximized, which implies an optimization problem of very high complexity to be solved.

For the solution of the optimization problem, we have explored an evolutionary approach [10]. To reduce the complexity almost to half, we have suggested a number of simple heuristics that exclude some combinations from consideration as superseded by others [11], and studied which heuristics give best results [12]. We have shown that, contrary to a popular belief, the Lesk algorithm, when proper optimization strategies are used, gives better results than so-called simplified Lesk algorithm [13].

We have also tried different similarity measures for this task [14], as well as suggested a number of similarity measures mentioned in Section III.C below. In particular, web search engines can provide important information on co-occurrence of words of the text with words from the context, which can be used for Lesk or the simplified Lesk algorithm, which does not require complex optimization [15].

McCarthy [16] suggested an excellent method for measuring the a priori probability for a word out of context to have a given sense. We have extended this method to directly measure the probability for a word to have a given sense in a given context [17]. Similarly to the simplified Lesk algorithm, our method does not require a complex optimization, but similarly to the original Lesk algorithm, it uses the information from sense definitions. The method uses two-stage strategy [18] and a dynamically built thesaurus [19].

B. Collocations and Lexical Functions: Word Sense Disambiguation without Word Senses

Apart from the meaning of single words, it is very important to study the meaning of word combinations, or collocations [20] (in this paper we for simplicity use these terms interchangeably). Examples of word combinations are: *loud voice*, *eat bread*, *dog barks*, *deeply regret*. There are two main reasons to study collocations: first, not all words can be naturally combined; second, the meaning of a word combination can differ from the combination of the meanings of the two words.

The knowledge of what words can be combined in a given language is important for language learning and thus needs to be specified in dictionaries [21]. In addition, there are numerous computational applications that require this knowledge [22]. Here we refer not to statistical language models (which are no doubt of greatest importance for NLP applications) but to linguistic knowledge; even a high-frequency phenomenon can be ungrammatical and thus unacceptable in formal language, and even a low frequency phenomenon can be perfectly correct.

One of interesting applications of the knowledge of common word combinations is predicting a second word given the first word [23]. Say, with a suitable dictionary [24] you can restore a missing part in an elliptical phrase in a dialogue: (waiter) "*We have wine, beer, and refreshments.*" – (client) "*Red or white?*" (meaning "*red wine or white wine?*") [25].

Another application of the knowledge of word combinations is the detection and correction of malapropisms, or real-word errors [26], such as in *Mine is a long and a sad tail!* (this was how Carroll's Alice understood the Mouse): here, *sad tail* is not a common word combination, while a substitution by a homophone *tale* gives a collocation present in the dictionary: *sad tale*. A dictionary that lists existing words written or pronounced similarly to a given one, called its paronyms, can accelerate the search and correction of such errors [27].

Yet another application is measuring text cohesion: a text is cohesive if its words form many collocations present in the dictionary. This permits, for example, automatic splitting of text into paragraphs: within a paragraph, words are semantically related, i.e., the text is cohesive; words from different paragraphs form fewer correct collocations. Thus, a good splitting point in the text is where words form many collocations with each other at either side of the boundary, but few collocations across the boundary [28].

There are a number of applications of a collocation dictionary in cross-lingual settings, for example, in translation aid and in teaching foreign languages [29]. Even a simple statistics of occurrences of words next to each other, which can be obtained with major Internet search engines, can be a great help in manual translation or in solving doubts about the choice of a word in writing [30].

Given that information about collocations is important, a natural question arises how to collect it.

They can be collected manually, or collected automatically from texts using a syntactic parser, especially a dependency parser [31]. These collocations can be used as seeds for automatically generating new possible collocations using ontological knowledge, for example, given a collocation *visit city* and knowing that Mysore is a city, a new collocation *visit Mysore* can be automatically predicted to be correct. This way a great number of collocations can be predicted even if they have never been observed in real texts [32]. Note that, again, the theoretical possibility of combining two words in the language is not the same as observed probability of their co-occurrence in a corpus.

As we have mentioned, a second reason to study word combinations is the fact that their meaning can be non-compositional. This phenomenon also occurs when more than two words are involved, and such compounds are generally known as multiword expressions [33]; however, here we will discuss only pairs of words.

There is a very important and broad class of word combinations, where one word retains its original meaning, but the other expresses a quite general meaning different from the usual meaning of this word: for example, in *strong coffee*, *high temperature*, *loud voice*, *violent storm*, *deep silence*, the meaning of the first word is essentially the same: 'very', high intensity. The phenomenon of the choice of different words to express the same meaning in such cases is studied in the theory of so called lexical functions [34].

A great part of ambiguity of word senses discussed in Section A, in fact, results from the word serving as a lexical function: many senses that appear in dictionaries, especially bilingual dictionaries, are in fact values of lexical functions: say, a usual translation of *strong* into Spanish is *fuerte* ('having force'), but dictionaries also have to give a translation *cargado* ('charged') only due to its occurrence as a lexical function in such expressions as *strong coffee*, in Spanish *café cargado*. However, there is no need to specify all such senses in dictionaries, because they can be automatically predicted—which as a byproduct greatly reduces the number of choices in the WSD task.

Namely, using the concept of lexical function, relationships between collocations can be established, a kind of proportion *violent* : *storm* = *deep* : *silence* [35]. Such proportions can be learnt automatically from examples, and used to predict the meaning of word combinations not present in the dictionary [36], for example, *deep stillness*. I am not a native English speaker and I was not taught this collocation, but if I see it, I will understand it as *stillness of high intensity*, basing on the semantic similarity between *silence* and *stillness*. We have developed learning algorithms to model this type of reasoning [37], using the WEKA package [38], and applied it to Spanish as a case study [39], for which we have developed a seed lexicon of lexical functions.

C. *Figurative Use, Modality and Negation*

As we have seen, the meaning of a word can be prescribed in the dictionary (even though ambiguously, which results in the WSD task), or can be determined from a lexical function of its syntactic neighbor. However, wider context can also dramatically change the meaning of a word.

One such case is figurative use, when a word is used in a totally different sense than any sense known for it in the language system: for example, "a piece of cake" can refer to something easy, not to a portion of food. This is a particularly difficult case because grammar and dictionaries give little clue to distinguish the literal use of a phrase from the figurative use, which is necessary for correct interpretation of the text. Statistical methods using distributional statistics can be used for this task [40].

Another alternation of meaning in the broad context, in contrast to the lexical meaning specified in dictionaries, is modality and negation (which can be though as a particular case of modality): for example, an event described in a text can be not a real event but desired, proposed, suspected, etc., or the text can specifically imply that the event does not take place, which is negation. In our experience so far, this can be best modeled using hand-crafted rules [41].

D. *Recognizing Textual Entailment: Doing Semantics without Doing Semantics*

Mapping the text to its meaning, including disambiguation, is required because the program should reason in terms of meanings and not texts. However, in many applications what one observes in the input and in the output of the system are texts. Logical operations on the meanings of these texts can be modeled by operations on the texts directly, which basically eliminates, or substitutes, the need of finding the exact meaning of both texts.

A most common pattern of such reasoning involves implication between the meanings: Does the meaning of a text T logically imply that of a hypothesis H—irrespective of what both of those meanings? For example, given a text *Google's recent acquisition of Waze did not affect the stock market* (T), do you think that *Google acquired Waze* (H)? From *John's assassin is in jail*, do you think that *John is dead*? From *Marie lives in France*, do you think that *Marie lives in Paris*?

Many tasks have been shown to be particular cases of this very general task called Recognizing Textual Entailment (RTE). For instance, in question answering (see Section IV.A), the answer should be implied by its supporting text (text where the answer is "found"); in information retrieval, the result and a part of the document should be equivalent, that is, imply each other. Thus the RTE task is considered by many to be an easy workaround to the complexity of semantic analysis, a workaround of behaviorist kind: we are interested in the relations between the observable inputs without looking "inside" them to understand what their meanings are and how they interact.

Given the importance of the task, international competitions are organized, in which our systems have participated [42]. In some of them, we have used machine learning [43] and statistics-based approach [44][45] for the RTE task. As features, we used, in particular, lexical and syntactic properties [46].

In other systems, we measured the similarity between the two texts [47], in particular semantic edit distance [48] and lexical and syntactic similarity [49], with the idea that if the texts are similar, they are likely to be logically equivalent or one of them imply the other. While counterexamples can easily be constructed, in naturally occurring texts it seems to be highly probable that when people use similar words, they speak about the same or related events.

A deeper analysis is more promising when subtle logical inference is involved. Dependency parsing helps us to reveal logical relationships within the two texts and map them between the texts [50][51]. Anaphora resolution permits to detect the identity of the concept when in one of the two texts it is referred by a noun and in another, by a pronoun [52], which makes the system more robust [53]. In an even more logically-oriented approach [54], we used a complex quasi-semantic graphical representation of text called Universal Networking Language [55].

As we see, the RTE task, being almost equivalent to semantic analysis of text, requires all the strength of computational methods, such as accurate parsers and expensive lexical resources—which are much easier available for English than for languages with which we worked, such as Hindi, Bengali, or Spanish. A simple workaround for this problem is automatic translation, which should in principle preserve meaning: translate all involved texts into English and solve the semantic problem there [56]; the resulting methodology is, thus, language-independent [57]. We used web-based translation engine for the RTE task [58].

In addition to the basic RTE task formulated here (deciding whether or not the text T implies the hypothesis H: T $\Rightarrow$ H), there can be variations of the task. For example, one can consider a multi-way decision: given T_1, T_2, decide whether $T_1 \Rightarrow T_2$, $T_1 \Leftarrow T_2$, $T_1 \Leftrightarrow T_2$, $\neg (T_1 \wedge T_2)$, i.e., T_1 and T_2 are incompatible, or T_1 and T_2 are independent [59]. Another variation is cross-lingual setting: when the two texts are in different languages [60]. This is important, for example, to automatically answer questions in, say, Spanish, using an English document collection as a knowledge source.

A common disadvantage of similarity-based methods is that they typically are symmetric: they treat the cases such as *Marie lives in France* $\Rightarrow$ *Marie lives in Paris* and *Marie lives in Paris* $\Rightarrow$ *Marie lives in France* in the same way, as equivalence (which gives a 50% success rate even when they are not really equivalent; not so bad). With a more detailed analysis, the non-symmetric nature of the task is to be taken into account. We used our Soft Cardinality similarity measure to specifically learn directional entailment relationships in cross-lingual setting [61].

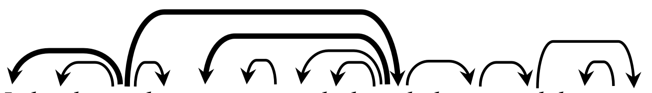
III. Representation and Similarity

Computational algorithms usually cannot work with the "text" in all complexity of its meaning and grammar, as humans perceive it. Instead, they work with some simplified *representation* of the text, which makes it simple process for a given purpose. Different representations are suitable for different tasks, often more complex representations being required for more complex tasks.

A. Syntactic Dependency n-grams

For the majority of machine learning methods, each object is to be represented as a vector, whose coordinates are, roughly speaking, the presence or absence of some features of the object. Most often as features of a text, individual words, pairs of adjacent words (bigrams), or more generally chains of *n* adjacent words called *n*-grams are used. This reflects the intuition that words that are adjacent in the text have good chances to be related and thus to form one concept, such as *eat bread* or *nice little girl*.

However, linguistic analysis can give a better idea of what words are related; words that are not adjacent in the linear order of the text can be "adjacent" in its meaning. In the following sentence:



I clearly see there a very nice little girl playing with her toys.

the words *see* and *girl* are "adjacent" in the meaning, as is clear from its dependency syntactic tree shown with the arrows. Examples of conventional *n*-grams in this sentence are: *there a very*, *nice little*, and *with her*. These words, however, don't combine in any meaningful idea. Thus we have proposed [62] to use sub-trees of the syntactic tree as features.

In our example, such syntactic tri-grams are *there* $\leftarrow$ *see* $\rightarrow$ *girl*, *see* $\rightarrow$ *girl* $\rightarrow$ *nice*, *playing* $\rightarrow$ *with* $\rightarrow$ *toys*, *a* $\leftarrow$ *girl* $\rightarrow$ *playing*, etc.; 4-grams are *playing* $\rightarrow$ *with* $\rightarrow$ *toys* $\rightarrow$ *her*, *I* $\leftarrow$ *see* $\rightarrow$ *girl* $\rightarrow$ *a*, etc. Here syntactic sub-graphs are shown; in plain English the latter one, shown with thick arrows in the picture, reads as *I see a girl*. Note that this is not a conventional *n*-gram. Syntactic *n*-grams convey much better meaning than conventional *n*-grams do, while it is equally easy to use it in feature vectors [63], in particular, in classification for NLP applications [64].

We also have explored the use of another generalization of the notion of *n*-gram, namely, maximal frequent sequences, which will be discussed in Section IV.B.

B. Conceptual Graphs

Conceptual graphs (CGs) suggested by J. Sowa are a rich semantic representation of the text's meaning. A CG is essentially a hyper-graph: a set of concept nodes such as objects, properties, or actions, and many-to-many relations (which Sowa calls relation nodes): *John is between Jill and Jane* is represented with three concept nodes: *John*, *Jill*, *Jane*, and one hyper-arc (relation node) outgoing from *John* and pointing to *Jill* and *Jane*. Rich semantics that can be represented with such structures and manipulated with well-defined operations can be used in numerous applications that require detailed understanding of text.

We have applied CGs to information retrieval [65], as well as to the task of finding semantic deviations in a collection of documents [66] and other tasks of text mining [67]. The use of

CGs allowed us to take into account fine details of the texts [68]. Applications of CGs in text summarization will be discussed in Section IV.B.

The key to such applications were our techniques for measuring similarity and other relationships (such as inclusion or entailment) between texts represented as CGs [69]. Such comparison can be made flexible, allowing the user fine control over what aspects are compared and how [70].

However, the task of constructing CGs from plain text is still far from being solved. On the other hand, not all aspects of the classical CGs are necessary for a particular application. We have worked on simplified versions of the CG formalism that would be easier to construct from plain text and still usable for practical tasks. One such form was CG-like knowledge representation scheme that we used for requirements elicitation from text in software technology [71]. Another one proved to be useful for information retrieval tasks [72].

C. *Text Similarity Measures*

As is the case with CG matching, in general measuring similarity between two texts is an underlying task of many other NLP methods and applications. This task is also of behaviorist nature: even if we don't really "understand" the text, much of our linguistic behavior (and of that of NLP algorithms) is based on what texts we deem mean the same, not what specifically they mean.

For example, in information retrieval, documents are retrieved based on their similarity to the query. Lesk-like WSD methods discussed in Section II.A rely on the similarity between the definitions of word senses. A number of approaches to RTE discussed in Section II.D are based on similarity between the text and the hypothesis.

A most common way of measuring similarity between two strings is the number of characters in common: the words *bread* and *broad* have 80% of letters in common. A more elaborate way of measuring the distance (the inverse of similarity) is the edit distance: the number of simple operations (such as letter substitutions, deletions, insertions, and transpositions) to be applied to one of the strings in order to convert it to the other. A particular case of an edit distance, which uses averaging of different kinds of simpler distances, is the Mongue-Elkan distance. We have shown that using weighted averaging instead of simple averaging improves the results of application of the Mongue-Elkan distance [73].

Usually the similarity is calculated as a quantity by a formula. However, we have shown that similarity values can also be though of as independent categories (in our case these were integer values from 1 to 5), and a classifier can be trained to classify string pairs into these categories [74]. As features, we used a large number of different classical similarity measures, so that our classifier was a kind of a voting scheme.

Abstract strings are usually compared considering all their letters in the same way. However, when we compare the meaning of words, the significance of letters in them generally depends on the letter's position: say, in many languages final letters are least significant for the word's meaning: *ask*, *asks*, *asked*, and *asking* refer to the same action despite the difference in the final letters, while *ark* is a totally different word. While the most accurate way to address the problem of identifying such words is full-fledged morphological analysis [75], very simple statistical considerations can be used to estimate the probability for two words to be morphological or derivational variants of each other [76].

On the other hand, for some applications such as spelling correction we might be interested in the opposite question: what words of a given language have different meaning but are "similar" enough to be confused in writing, speaking, or listening (such as *tail*—*tale* in the example in Section II.B). Different models of such errors (what and how people may confuse) lead to different similarity measures to identify such error-prone pairs: for example, the words *clearness*, *clearing*, *clearance*, **clearhood*, and **cleardom*, though they sound very different, for a non-native speaker may be equally plausible candidates to express the meaning 'gap, empty space' [77].

Similarity is usually thought to be a symmetric measure: *a* is as similar to *b* as *b* is to *a*. However, evidence exists that humans often perceive similarity in a non-symmetrical way: say, *Korea* is more similar to *China* than *China* is to *Korea*. This effect is important in such NLP applications as resolving variation in names of entities such as *US* vs. *USA* vs. *United States* [78].

For identifying semantic similarity between words and phrases, the analysis of distribution statistics is widely used. Such analysis requires very large corpora. However, dictionaries of word combinations (collocations) discussed in Section II.B can be used for measuring semantic similarity: say, if two nouns are naturally combined (that is, appear in a dictionary of collocations) with similar sets of adjectives and verbs, then these nouns are probably similar in meaning [79]. We have developed different methods for calculating such similarity [80].

To measure similarity between complete texts, not just words, one can use the fact that words, unlike letters, have meaning and thus can be mapped to an ontology. The frequencies of the words can be propagated up the hierarchy: say, one text 23 times mentioned animals (cows, lambs, and sheep), and the other text 22 times (cats, dogs, and parrots); even if specific animals are different, the texts are likely to be on similar topics [81]. Note that in our method the topics are not just categorized (animals, books, etc.) but organized in a deep tree, so that it is weighted trees what is compared.

As we have seen, in many cases similarity between two objects is measured as the size of intersection between these objects—for instance, as the number of common words in texts. Obviously, the size of a set is just the number of items in it... or is it? For a simple example, the words in the set {*guy*, *lad*, *chap*} refer to the same thing, so for many applications it is convenient to consider that there is only one word (concept) in this set and not three different ones. Or perhaps a little bit more than one word, given that they are still slightly different. Thus, two texts that have these words in common, or differ in these words, in fact have a little bit more than one word in common, or, correspondingly, differ in, say, 1.2 words. For less similar words the set can look bigger: {*dog*, *bird*, *fish*} can be considered somewhat less than three different words, say, 2.3. We called this measure of the "size" of the set soft cardinality [82].

Soft cardinality of a set can be approximately computed in linear time [83]. The specific measure depends in turn on how the similarity of individual words is measured and how pair-wise similarity values are combined into the soft cardinality value of the whole set, so parameterized versions of the soft cardinality exist [84].

Soft cardinality can be used in a very wide class of applications that implicitly or explicitly deal with sets and their sizes—which is perhaps the majority of applications. Similarly to how fuzzy logic implies a fuzzy version of any mathematical notion, soft cardinality can probably provide a soft version of nearly any NLP algorithm. We have used it for text comparison alone [85] or in combination with other methods already mentioned in this section, such as distributional analysis [86] and hierarchical overlap measure [87].

IV. Applications

The methods described above, as well as other methods, have been used in a number of applications, of which due to space limitations I can here briefly present only four main directions in which we have worked: question answering, text summarization, emotions and polarity, and music generation.

A. *Question Answering*

One application of NLP is well-known and used every day by all Internet users: information retrieval, or, simply said, Google. In information retrieval task the user formulates a query in the form of keywords and obtains a list of documents where these words are used. However, very often this is not what the user really wants to know; instead, the user has a question and wants to get a simple answer: yes or no, or who, when, where, how, instead of looking for this information in documents.

Technology that allows giving answers to such questions is called Question Answering (QA). Google has recently incorporated this technology in its search engine: the query *Where was Obama born?* results now in the answer *Honolulu, Hawaii, United States* and not just a long list of documents.

A typical QA system consists of two large modules: one that locates possible answer candidates in a large document collection, and one that evaluates each candidate in order to present to the user the best one. The former process is called answer generation and the latter, answer validation. We have explored various techniques for answer validation [88].

As we have noted in Section II.D, RTE technology provides good clues for answer validation: the answer should be implied by the supporting text (the text passage on the basis of which the answer generation module built it), that is, should be correct, as well as should imply the question in some re-phrased form, that is, answer this specific question [89].

Semantic reasoning is also an important approach to answer validation. While application of CGs (see Section III.B) is a possible direction of our future work, we have used a more practical, though less precise, representation of the text's semantics via UNL [55] (discussed in Section II.D), obtaining a semantic-based answer validation system [90].

Our answer validation systems have successfully participated in the international competitions at the Cross Language Evaluation Forum events [91], and one of them was the best one of the ten systems that competed in the Question Answering for Machine Reading Evaluation track [92].

Usually the answer to the question is contained in one single document, specifically in one single place: for example, for *Where was Obama born?*, Wikipedia states: "Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School". However, in some tasks it is important to combine information from different documents. Such is often the case of answering questions on legal documents such as laws, rules, and regulations: only reasoning that involves several different articles of a law, or even different laws, provides the answer. We used a representation of a law or a collection of laws as a graph, whose nodes are articles and arcs reflect relatedness o similarity between the articles, such as common concepts [93].

Then we generate the answer as a path in this graph. For example, suppose article (a) of the university regulations states that PhD exam jury should include an external examiner, article (b) states that the external examiner should be approved by the academic committee, and article (c), that the academic committee consists of the Dean and heads of the departments. The graph includes nodes (a), (b), and (c), and their intersections *external examiner* and *academic committee* are arcs. Then the answer to a question *Who assigns the jury for PhD exam?* includes the path (a)—(*external examiner*)—(b)—(*academic committee*)—(c).

NLP techniques, such as similarity measures discuss in Section III.C, are required for construction of the arcs of such a graph [91]. However, apart from plain text, the articles of a law or rules contain explicit mentions of other articles of the same or other applicable documents, much like web pages contain hyperlinks. Similarly to search engines, NLP approaches can be combined with, or substituted by, the analysis of explicit link structure [95].

B. *Text Summarization*

Text summarization is a task of generating a short document that reflects the important points of a longer document or a set of documents, such as all news for today or all documents found by a search engine given a user query. This saves the reader's time on familiarizing himself or herself with the document or collection of documents.

There are two main types of summaries: extractive summaries and abstractive summaries. In extractive summarization, the program composes the summary out of pieces of the text, much like what we do when we use a yellow marker to highlight the important places in the text. In contrast, in abstractive summarization the program constructs a new text, as we do when we re-tell a story in our own words.

One possible approach to extractive automatic summarization is to group the sentences of the document into clusters of sentences on similar topics, and choose one sentence (a representative one) from each group. In this way the most important topics of the document are covered in the generated summary. We have shown that the Expectation Maximization algorithm outperforms the *k*NN algorithm for the clustering step in this task [96].

To decide which sentences are most important (that is, which sentences to "highlight with yellow marker"), typically a measure of importance is assigned to single words, and then the importance of each sentence is calculated as the sum of the importance values of its words. Instead of individual words, we have suggested using larger units of text that are likely to convey specific meaning, namely, maximal frequent sequences (MFSs) [97]. An MFS is an *n*-gram that is often found in texts (and is not a part of a longer almost equally frequent *n*-gram). For example, *United States of America* is a frequent sequence and thus is likely to denote a concept (which it does: a country), while *United States of* is a part of an almost equally frequent but longer *n*-gram and thus is not an MFS. We have shown that using MFSs has advantages over using single words [98].

Abstractive summarization is a much more promising approach than extractive summarization because it allows for much better quality of the generated text, as well as for non-trivial generalization operations. For example, if the text contains passages about *dogs*, *cats*, and *parrots*, for extractive summaries one of them is to be chosen; in contrast, an abstractive summary can state that the text is about *mascots*. However, this approach is much more difficult to achieve. It typically relies on analysis of text, reducing its semantic representation, and generating a new text basing on that new semantic structure.

CGs (Section III.B) is a good semantic representation of text. We have developed methods for reducing the semantic representation of text in the form of a collection of CGs, using non-trivial operations such as generalization, joining, or pruning the CGs [99]. Our experiments show that if the complete processing is achieved (as I have mentioned, translation of plain text into CGs, as well as generation of plain text from CGs, is still an open question) then our system will outperform the current state-of-the-art methods by about 20%.

C. *Sentiment Analysis and Opinion Mining*

Analysis of emotions and opinions has recently attracted very strong attention from the industry and governments, which therefore, due to huge investments, sparked the interest to these topics in the research community. Nowadays nearly any major NLP group has sentiment analysis and opinion mining among his interests, if not among its first priorities.

Detecting emotions that the users express in blogs and social networks helps businesses to better understand the needs of the consumers and improve their products, helps the consumers to choose better products and services, and helps the governments to improve their projects. It therefore helps improving economy, quality of life, and democracy.

Technically, analysis of emotions in the text usually relies on lexical resources: dictionaries where associated emotions are specified for each word. We have built a large dictionary of emotions for English [100]. This dictionary is an extension of previously existing SenticNet dictionary [101], built in frame of the novel Sentic Computing paradigm [102].

The original SenticNet dictionary only specified polarity: whether a word invokes positive or negative emotions, without specifying the type of emotion. What we did was to automatically assign a specific emotion to each concept in SenticNet, using WordNet Affect, a small dictionary of emotions, as seed data [103]. This was achieved by semi-supervised clustering of the SenticNet concepts into categories corresponding to the six basic emotions [104]. For this, we have developed a novel algorithm of semi-supervised clustering and novel similarity measures for words [105].

We have also constructed an independent emotion lexicon for Spanish and applied it to the study of micro-blogs in Twitter [106]. This lexicon was constructed manually with the help of a large number of native speaking annotators; a novel procedure of combining their annotations was also developed.

D. *Music Generation and Classification*

While analysis of music lies outside the goals of NLP, some techniques originally developed for NLP prove to be useful in music applications.

One such case is the formal grammar formalisms. We have shown that even regular grammars can be used to learn and express interesting regularities in music composition. We trained such a grammar on human-composed musical pieces and then used a probabilistic process to generate music with similar spectrum [107].

In another set of experiments, we used more complex, context-dependent grammars to generate music. Our goal in those experiments was multi-instrument music: each instrument was modeled with its own grammar, whose parameters were adjusted to express this particular instrument (such as piano or bass). The main novel contribution of this work is the using of a "conductor" to coordinate those instruments for them to generate a coherent melody [108]. For this, each instrument's grammar uses a set of parameters that can be dynamically adjusted by an external agent. One additional grammar, the "conductor" of the orchestra, generates a sequence of parameters, which are then simultaneously set in each instrument's grammar, thus achieving coordination of the melodies played by different instruments.

Finally, the semi-supervised methods that we have developed for classification of words by emotions (Section C) proved to be useful in semi-supervised classification of music pieces by genre [109]. We believe that those methods are general enough to be useful in many different applications, and plan to test this in our future work.

ACKNOWLEDGMENTS

The work was partially supported by the Government of Mexico (CONACYT 122030; SIP-IPN 20131702 and 20131441), Government of Mexico City (ICYT PICCO10-120), and Government of India (CONACYT-DST project "Answer validation through textual entailment"), and Mexican Society of Artificial Intelligence (SMIA). The author is deeply

International Conference on Advances in Computing, Communications and Informatics (ICACCI 2013), ISBN 978-1-4673-6217-7, IEEE Conference Publications, pp 2112–2121, [illegible]

grateful to the Organizing Committee of the ICACCI 2013 and the Sri Jayachamarajendra College of Engineering for the invitation to present a keynote address at the conference.

References

[1] Y. Ledeneva, G. Sidorov. "Recent Advances in Computational Linguistics," Informatica, Vol. 34, 2010, pp. 3–18.

[2] G. Sidorov, J.-P. Posadas-Durán, H. Jiménez Salazar, L. Chanona-Hernandez. "A new combined lexical and statistical based sentence level alignment algorithm for parallel texts," International Journal of Computational Linguistics and Applications, vol. 2, no. 1–2, 2011, pp. 257–263.

[3] N. A. Castro-Sánchez, G. Sidorov. "Automatic Acquisition of Synonyms of Verbs from an Explanatory Dictionary using Hyponym and Hyperonym Relations," Lecture Notes in Computer Science, N 6718, 2011, pp. 322–331.

[4] E. Agirre, P. Edmonds (Eds.) Word Sense Disambiguation: Algorithms and Applications. Springer, 2006.

[5] A. Gelbukh, G. Sidorov, Y. Ledo-Mezquita. "On similarity of word senses in explanatory dictionaries," International Journal of Translation, Vol.15, No. 2, 2003, pp. 51–60.

[6] A. Gelbukh, G. Sidorov, S. Han, L. Chanona-Hernandez. "Automatic Evaluation of Quality of an Explanatory Dictionary by Comparison of Word Senses," Lecture Notes in Computer Science, N 2890, 2003, pp. 556–562.

[7] H. Saarikoski, S. Legrand, A. Gelbukh. "Case-Sensitivity of Classifiers for WSD: Complex Systems Disambiguate Tough Words Better," CICLing 2007, Lecture Notes in Computer Science, N 4394, 2007, p. 253–266.

[8] H. Saarikoski, S. Legrand, A. Gelbukh. "Defining Classifier Regions for WSD Ensembles Using Word Space Features," Lecture Notes in Artificial Intelligence, N 4139, 2006.

[9] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone." 5th SIGDOC, New York, 1986, pp. 24–26.

[10] A. Gelbukh, G. Sidorov, S. Han. "Evolutionary Approach to Natural Language Word Sense Disambiguation through Global Coherence Optimization," WSEAS Transactions on Communications, Issue 1 Vol. 2, 2003, p. 11–19.

[11] A. Gelbukh, G. Sidorov, S. Han. "On Some Optimization Heuristics for Lesk-Like WSD Algorithms," Lecture Notes in Computer Science, N 3513, 2005, pp. 402–405.

[12] A. Gelbukh, S. Han, G. Sidorov. "Comparison of some global coherence optimization heuristics for word sense disambiguation," CIC-2003, XII Congreso Internacional de Computación, p. 131–135.

[13] S. Torres. "Optimización global de coherencia en la desambiguación del sentido de las palabras," PhD thesis, IPN, Mexico, 2010.

[14] S. Torres and A. Gelbukh. "Comparing Similarity Measures for Original WSD Lesk Algorithm," Research in Computing Science, N 43, pp.155–166.

[15] M. Á. Ríos Gaona, A. Gelbukh, S. Bandyopadhyay. "Web-based Variant of the Lesk Approach to Word Sense Disambiguation," MICAI 2009. Proc. of 2009 Eighth Mexican International Conference on Artificial Intelligence, IEEE CS Press, 2009, pp. 103–107.

[16] D. McCarthy, R. Koeling, J. Weeds, J, Carroll. "Finding predominant senses in untagged text," ACL 2004, Barcelona, Spain, 2004, pp 280–287.

[17] J. Tejada-Cárcamo, H. Calvo, A. Gelbukh, K. Hara. "Unsupervised WSD by Finding the Predominant Sense Using Context as a Dynamic Thesaurus. Journal of Computer Science and Technology," Vol. 25, No. 5, 2010, p. 1030–1039.

[18] J. Tejada-Cárcamo, A. Gelbukh, H. Calvo. "An innovative two-stage WSD unsupervised method," Procesamiento de Lenguaje Natural, N 40, 2008, pp. 99–106.

[19] J. Tejada-Cárcamo, H. Calvo, A. Gelbukh. "Improving Unsupervised WSD with a Dynamic Thesaurus," Lecture Notes in Artificial Intelligence, N 5246, 2008, pp. 201–210.

[20] I. A. Bolshakov, A. Gelbukh. "A Very Large Database of Collocations and Semantic Links," Lecture Notes in Computer Science N 1959, 2001, pp. 103–114.

[21] I. A. Bolshakov, A. Gelbukh. "A Very Large Dictionary with Paradigmatic, Syntagmatic, and Paronymic Links between Entries," International Workshop on Enhancing and Using Electronic Dictionaries at COLING 2004, Geneva, Switzerland, 2004, pp. 54–57.

[22] I. A. Bolshakov, S. N. Galicia-Haro, A. Gelbukh. "Stable Coordinated Pairs in Text Processing," Lecture Notes in Artificial Intelligence, N 2807, 2003, pp. 27–34.

[23] A. Gelbukh, G. Sidorov, I. Bolshakov. "On Coherence Maintenance in Human-Machine Dialogue with Contextual Ellipses," Computación y Sistemas, Vol. 5, N 3, 2002, pp. 204–214.

[24] A. Gelbukh, G. Sidorov, I. A. Bolshakov. "Dictionary-based Method for Coherence Maintenance in Man-Machine Dialogue with Indirect Antecedents and Ellipses," Lecture Notes in Artificial Intelligence, N 1902, 2000, pp. 357–362.

[25] A. Gelbukh, G. Sidorov, I. A. Bolshakov. "Coherence Maintenance in Human-Machine Dialogue: Indirect Antecedents and Ellipses," Proc. DEXA-2000, 11th International Conference and Workshop on Database and Expert Systems Applications, NLIS-2000, 2nd International Workshop on Natural Language and Information Systems, England, 2000, IEEE Computer Society Press, pp. 96–100.

[26] I. Bolshakov, A. Gelbukh. "On Detection of Malapropisms by Multistage Collocation Testing," Lecture Notes in Informatics, Bonner Köllen Verlag, 2003, pp. 28–41.

[27] A. Gelbukh, I. Bolshakov. "On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms," Lecture Notes in Artificial Intelligence, N 3034, 2004, pp. 105–114.

[28] I. A. Bolshakov, A. Gelbukh. "Text segmentation into paragraphs based on local text cohesion," Lecture Notes in Artificial Intelligence N 2166, 2001, pp. 158–166.

[29] I. A. Bolshakov, A. Gelbukh. "A Large Database of Collocations and Semantic References: Interlingual Applications," International Journal of Translation, Vol.13, No.1–2, 2001, pp. 167–187.

[30] A. Gelbukh, I. A. Bolshakov. "Internet, a true friend of translator: the Google wildcard operator," International Journal of Translation, Vol. 18, No. 1–2, 2006, pp. 41–48.

[31] A. Gelbukh, G. Sidorov, S. Han, E. Hernandez-Rubio. "Automatic Enrichment of a Very Large Dictionary of Word Combinations on the Basis of Dependency Formalism," Lecture Notes in Artificial Intelligence, N 2972, 2004, pp. 430–437.

[32] I. A. Bolshakov, A. Gelbukh. "Heuristics-Based Replenishment of Collocation Databases," Lecture Notes in Artificial Intelligence, N 2389, 2002, p. 25–32.

[33] A. Gelbukh, O. Kolesnikova. "Multiword Expressions in NLP: General Survey and a Special Case of Verb-Noun Constructions," In: S. Bandyopadhyay, Sudip Kumar Naskar, Asif Ekbal (eds.). Emerging Applications of Natural Language Processing: Concepts and New Research. IGI Global, pp. 1–21.

[34] A. Gelbukh, O. Kolesnikova. Semantic Analysis of Verbal Collocations with Lexical Functions. Studies in Computational Intelligence N 414, 2013, 146 pp.

[35] O. Kolesnikova, A. Gelbukh. "Semantic relations between collocations—A Spanish case study," Revista Signos, Vol. 45, No. 78, 2012, pp. 44–59.

[36] A. Gelbukh, O. Kolesnikova. "Supervised Machine Learning for Predicting the Meaning of Verb-Noun Combinations in Spanish. Olga Kolesnikova," Lecture Notes in Artificial Intelligence N 6438, 2010, pp. 196–207.

[37] A. Gelbukh, O. Kolesnikova. "Supervised Learning Algorithms Evaluation on Recognizing Semantic Types of Spanish Verb-Noun Collocations," Computación y Sistemas, Vol. 16, No. 3, 2012, pp. 297–308.

[38] O. Kolesnikova, A. Gelbukh. "Using WEKA for Semantic Classification of Spanish Verb-Noun Collocations," Research in Computing Science, N 45, 2010.

[39] A. Gelbukh, O. Kolesnikova. "Supervised Learning for Semantic Classification of Spanish Collocations," Lecture Notes in Computer Science N 6256, 2010, pp. 362–371.

[40] S. Jimenez, C. Becerra, A. Gelbukh. "UNAL: Discriminating between Literal and Figurative Phrasal Usage Using Distributional Statistics and POS tags," *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Vol. 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), USA. Collocated with NAACL 2013. The Association for Computational Linguistics, 2013, pp. 114–117.

[41] P. Pakray, P. Bhaskar, S. Banerjee, S. Bandyopadhyay, A. Gelbukh. "An Automatic System for Modality and Negation Detection," CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, track: Question Answering for Machine Reading Evaluation (QA4MRE), Italy, 2012.

[42] P. Pakray, Santanu Pal, Soujanya Poria, S. Bandyopadhyay, A. Gelbukh. "JU_CSE_TAC: Textual Entailment Recognition System at TAC RTE-6. System Report," Text Analysis Conference, Recognizing Textual Entailment Track (TAC RTE). Notebook, 2010.

[43] M. A. Ríos Gaona, A. Gelbukh, S. Bandyopadhyay. "Recognizing Textual Entailment Using a Machine Learning Approach," Lecture Notes in Artificial Intelligence N 6438, 2010, pp. 177–185.

[44] P. Pakray, U. Barman, S. Bandyopadhyay, A. Gelbukh. "A Statistics-Based Semantic Textual Entailment System," Lecture Notes in Artificial Intelligence N 7094, 2011, p. 267–276.

[45] M. Á. Ríos Gaona, A. Gelbukh, S. Bandyopadhyay. "Recognizing Textual Entailment with Statistical Methods," Lecture Notes in Computer Science N 6256, 2010, pp. 372–381.

[46] P. Pakray, S. Bandyopadhyay, A. Gelbukh. "A hybrid textual entailment system using lexical and syntactic features," 9th IEEE International Conference on Cognitive Informatics, ICCI 2010, China, IEEE CS Press, 2010, p. 291–296.

[47] M. Rios, A. Gelbukh. "Recognizing Textual Entailment with Similarity Metrics," Research in Computing Science, Vol. 58, 2012, pp. 337–347.

[48] M. Rios, A. Gelbukh. "Recognizing Textual Entailment with a Semantic Edit Distance Metric," 11th Mexican International Conference on Artificial Intelligence (MICAI), IEEE CS Press, 2012, pp. 15–20.

[49] P. Pakray, S. Bandyopadhyay, A. Gelbukh. "Textual Entailment using Lexical and Syntactic Similarity," International Journal of Artificial Intelligence and Applications, Vol.2, No.1, 2011, pp. 43–58.

[50] P. Pakray, A. Gelbukh, S. Bandyopadhyay. "A Syntactic Textual Entailment System Based on Dependency Parser," CICLing 2010. Lecture Notes in Computer Science N 6008, 2010, pp. 269–278.

[51] P. Pakray, S. Bandyopadhyay, A. Gelbukh. "Dependency Parser Based Textual Entailment System," Proc. of the 2010 International Conference on Artificial Intelligence and Computational Intelligence, AICI'10, China, IEEE CS Press, 2010, Vol. 1, p. 393–397.

[52] P. Pakray, S. Bandyopadhyay, A. Gelbukh. "Textual Entailment and anaphora resolution," 3rd International Conference on Advanced Computer Theory and Engineering, ICACTE 2010, China, IEEE CS Press, 2010, Vol. 6, 334–336.

[53] P. Pakray, S. Neogi, P. Bhaskar, S. Poria, S. Bandyopadhyay, A. Gelbukh. "A Textual Entailment System using Anaphora Resolution. System Report," Text Analysis Conference, Recognizing Textual Entailment Track (TAC RTE). Notebook. National Institute of Standards and Technology, USA, 2011.

[54] P. Pakray, S. Poria, S. Bandyopadhyay, A. Gelbukh. "Semantic Textual Entailment Recognition using UNL," Polibits, Issue 43, 2011, p. 23–27.

[55] J. Cardeñosa, A. Gelbukh, E. Tovar (eds.) Universal Networking Language: Advances in Theory and Applications. Special issue of Research on Computing Science, Vol. 12, 2005, 443 pp.

[56] P. Pakray, S. Neogi, S. Bandyopadhyay, A. Gelbukh. "Recognizing Textual Entailment in Non-English Text via Automatic Translation into English," Lecture Notes in Artificial Intelligence, N 7630, 2013.

[57] S. Neogi, P. Pakray, S. Bandyopadhyay, A. Gelbukh. "JU_CSE_NLP: Language Independent Cross-lingual Textual Entailment System," Proc. of *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. Collocated with NAACL-HLT 2012, Canada, Association for Computational Linguistics; pp. 689–695.

[58] P. Pakray, Snehasis Neogi, S. Bandyopadhyay, A. Gelbukh. "A Textual Entailment System using Web based Machine Translation System," NTCIR-9: The 9th NTCIR Workshop Meeting "Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access". RITE competition: Recognizing Inference in TExt@NTCIR9. National Institute of Informatics (NII), National Center of Sciences, Tokyo, Japan, 2011.

[59] P. Pakray, S. Bandyopadhyay, A. Gelbukh. "Binary-class and Multi-class based Textual Entailment System," Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, Japan. National Institute of Informatics, 2013.

[60] S. Jimenez, C. Becerra, A. Gelbukh. "Soft Cardinality + ML: Learning Adaptive Similarity Functions for Cross-lingual Textual Entailment," Proc. of *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. Collocated with NAACL-HLT 2012. Canada. Association for Computational Linguistics, 2012, pp. 684–688.

[61] S. Jimenez, C. Becerra, A. Gelbukh. "SOFTCARDINALITY: Learning to Identify Directional Cross-Lingual Entailment from Cardinalities and SMT," *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Vol. 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). USA. Collocated with NAACL 2013. The Association for Computational Linguistics, 2013, pp. 34–38.

[62] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández. "Syntactic N-grams as Machine Learning Features for Natural Language Processing", Expert Systems with Application, 2013, in press.

[63] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández. "Syntactic Dependency-based N-grams as Classification Features," Lecture Notes in Artificial Intelligence, N 7630, 2013, pp. 1–11.

[64] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández. "Syntactic Dependency-Based N-grams: More Evidence of Usefulness in Classification," CICLing 2013. Lecture Notes in Computer Science N 7816, 2013, pp. 13–24.

[65] M. Montes-y-Gómez, A. López-López, A. Gelbukh. "Information Retrieval with Conceptual Graph Matching," Lecture Notes in Computer Science, N 1873, 2000, pp. 312–321.

[66] M. Montes-y-Gómez, A. Gelbukh, A. López-López. "Detecting deviations in text collections: An approach using conceptual graphs," Lecture Notes in Artificial Intelligence N 2313, 2002, pp. 176–184.

[67] M. Montes y Gómez, A. Gelbukh, A. López López, R. Baeza-Yates. "Text mining with conceptual graphs," IEEE International Workshop on Natural Language Processing and Knowledge Engineering, NLPKE 2001 at International IEEE SMC-2001 Conference: Systems, Man, And Cybernetics. USA, 2001, IEEE, pp. 898–903.

[68] M. Montes-y-Gómez, A. Gelbukh, A. López-López. "Text Mining at Detail Level Using Conceptual Graphs," Lecture Notes in Artificial Intelligence, N 2393, 2002, pp. 122–136.

[69] M. Montes-y-Gómez, A. Gelbukh, A. López-López. "Comparison of Conceptual Graphs," Lecture Notes in Artificial Intelligence, N 1793, 2000, pp. 548–556.

[70] M. Montes y Gómez, A. Gelbukh, A. López López, R. Baeza-Yates. "Flexible Comparison of Conceptual Graphs," Lecture Notes in Computer Science N 2113, 2001, pp. 102–111.

[71] C. M. Zapata Jaramillo, A. Gelbukh, F. Arango Isaza. "Pre-conceptual Schema: A Conceptual-Graph-like Knowledge Representation for Requirements Elicitation," Lecture Notes in Artificial Intelligence, N 4293, 2006, pp. 17–29.

[72] S. Ordoñez, A. Gelbukh. "Information Retrieval with a Simplified Conceptual Graph-like Representation," Lecture Notes in Artificial Intelligence, N 6437, 2010, pp. 92–104.

[73] S. Jimenez, C. Becerra, A. Gelbukh, F. Gonzalez. "Generalized Mongue-Elkan Method for Approximate Text String Comparison," CICLing 2009. Lecture Notes in Computer Science N 5449, 2009, pp. 559–570.

[74] S. Neogi, P. Pakray, S. Bandyopadhyay, A. Gelbukh. "JU_CSE_NLP: Multi-grade Classification of Semantic Similarity between Text Pairs," *SEM 2012: The First Joint Conference on Lexical and Computational

Semantics. Collocated with NAACL-HLT 2012. Canada, Association for Computational Linguistics, pp. 571–574.

[75] A. Gelbukh, G. Sidorov. “Morphological Analysis of Inflective Languages through Generation,” Procesamiento de Lenguaje Natural, No 29, 2002, p. 105–112.

[76] X. Blanco, M. Alexandrov, A. Gelbukh. “Modified Makagonov’s Method for Testing Word Similarity and its Application to Constructing Word Frequency Lists,” Research in Computing Science, Vol. 18, 2006, pp. 27–36.

[77] I. A. Bolshakov, A. Gelbukh. “Paronyms for Accelerated Correction of Semantic Errors,” International Journal on Information Theories and Applications, Vol.10, 2003, pp. 11–19.

[78] S. Jiménez, A. Gelbukh. “Human Factors for Assessing the Similarity of Texts in Entity Resolution,” Doctoral Consortium at the 10th Mexican International Conference on Artificial Intelligence, MICAI-2011, Mexico.

[79] I. Bolshakov, A. Gelbukh. “Distribution-Based Semantic Similarity of Nouns,” Lecture Notes in Computer Science, N 4756, 2007, p. 704–713.

[80] I. A. Bolshakov, A. Gelbukh. “Two Methods of Evaluation of Semantic Similarity of Nouns Based on Their Modifier Sets,” Lecture Notes in Computer Science, N 4592, 2007, p. 414–419.

[81] A. Gelbukh, G. Sidorov, A. Guzmán-Arenas. “Document comparison with a weighted topic hierarchy,” Proc. DEXA-99, 10th International Conference and Workshop on Database and Expert Systems Applications, DAUDD’99, 1st International Workshop on Document Analysis and Understanding for Document Databases, Italy, 1999, IEEE Computer Society Press, pp. 566–570.

[82] S. Jimenez Vargas, A. Gelbukh. “Baselines for Natural Language Processing Tasks Based on Soft Cardinality Spectra,” International Journal of Applied and Computational Mathematics, Vol. 11, No. 2, 2012, pp. 180–199.

[83] S. Jimenez Vargas, A. Gelbukh. “SC spectra: A linear-time soft cardinality approximation for text comparison,” Lecture Notes in Artificial Intelligence N 7095, 2011, pp. 213–224.

[84] S. Jimenez, C. Becerra, A. Gelbukh. “Soft Cardinality: A Parameterized Similarity Function for Text Comparison,” *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. Collocated with NAACL-HLT 2012, Canada, Association for Computational Linguistics, pp. 449–453.

[85] S. Jiménez Vargas, F. A. González O., A. Gelbukh. “Text Comparison Using Soft Cardinality,” Lecture Notes in Computer Science N 6393, 2010, pp. 297–302.

[86] S. Jimenez, C. Becerra, A. Gelbukh. “SOFTCARDINALITY-CORE: Improving Text Overlap with Distributional Measures for Semantic Textual Similarity,” *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Vol. 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity. USA. Collocated with NAACL 2013. The Association for Computational Linguistics, pp. 194–201.

[87] S. Jimenez, C. Becerra, A. Gelbukh. “SOFTCARDINALITY: Hierarchical Text Overlap for Student Response Analysis,” *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Vol. 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). USA. Collocated with NAACL 2013. The Association for Computational Linguistics, pp. 280–284.

[88] P. Pakray, S. Pal, S. Bandyopadhyay, A. Gelbukh. “Automatic Answer Validation System on English language,” 3rd International Conference on Advanced Computer Theory and Engineering, ICACTE 2010, China, IEEE CS Press, 2010, Vol. 6, p. 329–333.

[89] P. Pakray, S. Bandyopadhyay, A. Gelbukh. “Answer Validation using Textual Entailment,” CICLing 2011, Lecture Notes in Computer Science N 6609, 2011, pp. 359–364.

[90] P. Pakray, U. Barman, S. Bandyopadhyay, A. Gelbukh. “Semantic Answer Validation using Universal Networking Language,” International Journal of Computer Science and Information Technologies, Vol. 3, issue 4, 2012, pp. 4927–4932.

[91] P. Bhaskar, P. Pakray, S. Banerjee, S. Banerjee, S. Bandyopadhyay, A. Gelbukh. “Question Answering System for QA4MRE@CLEF 2012,” CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, track: Question Answering for Machine Reading Evaluation (QA4MRE). CLEF 2012, Italy, 2012, 12 pp.

[92] P. Pakray, P. Bhaskar, S. Banerjee, B. Chandra Pal, S. Bandyopadhyay, A. Gelbukh. “A Hybrid Question Answering System based on Information Retrieval and Answer Validation,” CLEF 2011 Workshop on Question Answering for Machine Reading Evaluation (QA4MRE). CLEF 2011 Labs and Workshop. 19–22 September, Amsterdam. Notebook Papers, 2011, 16 pp.

[93] A. Monroy, H. Calvo, A. Gelbukh. “Using Graphs for Shallow Question Answering on Legal Documents,” Lecture Notes in Artificial Intelligence N 5317, 2008, pp. 165–173.

[94] A. Monroy, H. Calvo, A. Gelbukh. “NLP for Shallow Question Answering of Legal Documents Using Graphs,” CICLing 2009. Lecture Notes in Computer Science N 5449, 2009, pp. 498–508.

[95] A. L. Monroy, H. Calvo, A. Gelbukh, G. G. Pacheco. “Link Analysis for Representing and Retrieving Legal Information,” CICLing 2013. Lecture Notes in Computer Science N 7816, 2013, pp. 380–393.

[96] Y. Ledeneva, R. A. García Hernández, R. Montiel Soto, J. R. Cruz Reyes, A. Gelbukh. “EM Clustering Algorithm for Automatic Text Summarization,” Lecture Notes in Artificial Intelligence N 7094, 2011, pp. 305–315.

[97] Y. Ledeneva, A. Gelbukh, R. A. García-Hernández. “Terms Derived from Frequent Sequences for Extractive Text Summarization,” CICLing 2008, Lecture Notes in Computer Science N 4919, pp. 593–604.

[98] Y. Ledeneva, A. Gelbukh, R. García Hernandez. “Keeping Maximal Frequent Sequences Facilitates Extractive Summarization,” Research in Computing Science, vol. 34, 2008, pp. 163–174.

[99] S. Miranda-Jiménez, A. Gelbukh, G. Sidorov. “Summarizing Conceptual Graphs for Automatic Summarization Task,” Lecture Notes in Computer Science, N 7735, 2013, pp. 245–253.

[100] S. Poria, A. Gelbukh, A. Hussain, D. Das, S. Bandyopadhyay. “Enhanced SenticNet with Affective Labels for Concept-based Opinion Mining,” IEEE Intelligent Systems, vol. 28, issue 2, 2013, pp. 31–38.

[101] E. Cambria, R. Speer, C. Havasi, A. Hussain. “SenticNet: A publicly available semantic resource for opinion mining,” In: Proc. of AAAI CSK, 2010, pp. 14–18.

[102] E. Cambria, A. Hussain. Sentic computing: Techniques, tools, and applications. Dordrecht, Netherlands: Springer, 2012, 153 pp.

[103] S. Poria, A. Gelbukh, E. Cambria, P. Yang, A. Hussain, T. Durrani. “Merging SenticNet and WordNet-Affect emotion lists for sentiment analysis,” 2012 IEEE 11th International Conference on Signal Processing, IEEE ICSP 2012. China, 2012. Vol. 2, pp. 1251–1255.

[104] S. Poria, A. Gelbukh, E. Cambria, D. Das, S. Bandyopadhyay. “Enriching SenticNet Polarity Scores through Semi-Supervised Fuzzy Clustering. Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction,” SENTIRE 2012, IEEE 12th International Conference on Data Mining Workshops (ICDMW), Belgium. IEEE CS Press, 2012, pp. 709–716.

[105] S. Poria, A. Gelbukh, D. Das, S. Bandyopadhyay. “Fuzzy Clustering for Semi-Supervised Learning—Case study: Construction of an Emotion Lexicon,” Lecture Notes in Artificial Intelligence, N 7629, 2013, pp. 73–86.

[106] G. Sidorov, S. Miranda-Jiménez, F. Viveros-Jiménez, A. Gelbukh, N. Castro-Sánchez, F. Velásquez, I. Díaz-Rangel, S. Suárez-Guerra, A. Treviño, J. Gordon. “Empirical Study of Machine Learning Based Approach for Opinion Mining in Tweets,” Lecture Notes in Artificial Intelligence, N 7629, 2013, pp. 1–14.

[107] A. García, A. Gelbukh, H. Calvo. “Music composition based on linguistic approach,” Lecture Notes in Artificial Intelligence, N 6437, 2010, pp. 117–128.

[108] R. G. Ramírez Moreno, A. Gelbukh. “Grammar-based Multiple-Instrument Music Generation with a Conductor,” submitted.

[109] S. Poria, A. Gelbukh, A. Hussain, S. Bandyopadhyay, N. Howard. “Music Genre Classification: A Semi-supervised Approach,” Lecture Notes in Computer Science, N 7914, 2013, pp. 254–263.