

Recognizing Textual Entailment with a Semantic Edit Distance Metric

Miguel Rios

University of Wolverhampton,
Research Group in Computational Linguistics,
Stafford Street, Wolverhampton, WV1 1SB, UK
M.Rios@wlv.ac.uk

Alexander Gelbukh

Center for Computing Research,
National Polytechnic Institute,
Mexico City, Mexico
www.gelbukh.com

Abstract—We present a Recognizing Textual Entailment (RTE) system based on different similarity metrics. The metrics used are string-based metrics and the Semantic Edit Distance Metric, which is proposed in this paper to address limitations of known semantic-based metrics and to support the decisions made by a simple method based on lexical similarity metrics. We add the scores of the metrics as features for a machine learning algorithm. The performance of our system is comparable with the average performance of the Recognizing Textual Entailment Challenges, though lower than that of the state-of-the-art methods.

Keywords-Natural Language Processing, Recognizing Textual Entailment, Machine Learning.

I. INTRODUCTION

The Recognizing Textual Entailment (RTE) task consists in deciding, given two text expressions, whether the meaning of one text is entailed from the meaning of the other text [1]. The RTE Challenge is a generic task which addresses common semantic inference needs across Natural Language Processing (NLP) applications.

Examples of Natural Language Processing (NLP) applications that have strong links to RTE are: Summarization, a summary should be entailed by the source text, and in Machine Translation Evaluation, the meaning of the system translation should entail the meaning of a human reference translation.

In order to address the task of RTE, different methods have been proposed and most of these methods rely on Machine Learning (ML) algorithms trained with similarity scores as features. For example, The Text and Hypothesis (T-H) pair is represented by a linguistic level annotation (e.g. syntax trees), and a series of transformations are applied to transform T into H. The motivation is that a few amount transformations between a T-H pair results into a positive entailment relation. The amount of transformations become a distance score which is used a feature to train the classifier. However, the transformation approaches usually tackle the syntactic phenomena in RTE [2].

In this work we describe an RTE system based on different similarity metrics, where the metrics are: simple string similarity metrics and a semantic metric. In addition, we propose a new Semantic Edit Distance metric. The goal of our

Semantic Edit Distance metric is to measure the differences between a T-H pair at a predicate-argument level. We use a back-off score based on syntactic information in case the semantic metric is not able to give a score for a given T-H pair. For the simple string similarity metrics we propose a feature selection method to chose the best combination of metric-representation pairs. We add the scores of the metrics as features into a ML method for RTE. Then, we compare our results with related work on RTE. The performance of our system is comparable with the average performance of the RTE Challenges, but the performance is lower with both the related work and the best methods.

In the rest of this paper we discuss the related work (Section II), present our proposed method (Section III), and compare its performance with previous work (Section IV). Finally, we give the conclusions and discuss the possible future work (Section V).

II. RELATED WORK

The motivation behind the transformation approach to RTE is that the entailment relations can be measured by applying a series of transformations over T into H. This means that if the cost of a series of transformations over T is low, T is similar to H and they hold an entailment relation.

Edit distance algorithms are a common approach to transform texts. Where the basic operations to edit are: insertion, substitution and deletion. So, each operation has an attached score, this means that some operations are more expensive than others, and this cost is usually learned via ML algorithms. Then, the edit distance algorithms score the difference between a pair of texts based on how many operations were necessary to transform one text into another text. Kouylekov and Magnini [2] introduce the Edit distance algorithms for RTE. The assumption is based on estimating the cost of the information of the hypothesis which is missing in the text. The T-H pair holds an entailment relation if there is a sequence of operations over T such that we can produce H with an overall cost below a certain threshold. The threshold as well as the cost of each operation are learned from the development dataset by using ML techniques.

Moreover, Cabrio *et al.* [3] describe a system framework that consists of a combination of specialized entailment engines each addressing a specific entailment phenomenon. Because RTE is a combination of several phenomena which interact in a complex way. So, each engine is trained to deal with a different aspect of language variability (e.g. syntax, negation, modal verbs). Also, this framework has a modular approach to evaluate the progress on a single aspect of entailment using the training data. Basically the entailment engines are based on edit distance algorithms. In each engine the cost of each edit operation is defined (learned) according to a specific phenomenon. The cost schemes of the different engines are defined in order not to intersect each other. If the costs of the edit operations are set as not 0 for a certain phenomena, they are set as 0 for the aspects that are considered by another engine.

Transformation approaches can be combined with ML standard techniques and other approaches. Roth and Sammons [4] use semantic logical inferences for RTE, where the representation method is a Bag-of-Lexical-Items (BoLI). The BoLI relies in word overlap, in which an entailment relation holds if the overlap score is above a certain threshold. An extended set of stop words is used to select the most important concepts for the BoLI (auxiliary verbs, articles, exclamations, discourse markers and words in WordNet). Also, in order to recognize relations over the T-H pairs the system checks matchings between Semantic Role Labels (SRL), and then applies a series of transformations over the semantic representations to make easier to determine the entailment. The transformation operations are: *annotate* make some implicit property of the meaning of the sentence explicit. *Simplify/Transform* remove or alter some section of T in order to improve annotation accuracy or make it more similar to H. *Compare* (some elements of) the two members of the entailment pair and assign a score that correlates to how successfully (those elements of) the Hs can be subsumed by T.

On the other hand, Burchardt *et al.* [5] introduce new features for RTE. The new features as well as other methods involve deep linguistic analysis and shallow word overlap. The method consists of three steps: first, represent the T-H pair with the Frame Semantics (FS) and Lexical Functional Grammars (LFG) formalisms (the representation is similar to Semantic Role Labeling). Second, extract a similarity score based on matching the LFG graphs, and then make a statistical entailment decision. Burchardt *et al.* [5] use the RTE-2 and RTE-3 datasets as training data, and 47 features are extracted from the deep and the shallow overlap. The features consist of combinations of: predicates overlaps, grammatical functions match and lexical overlaps.

The methods which use Semantic Role Labeling (SRL) for RTE use the annotation provided by a semantic parser to measure the similarity between texts, but only measure the similarity in terms of how many labels they share (overlaps)

and not the content of those labels. Delmonte *et al.* [6] introduced semantic-mismatch features such as: locations, discourse markers, quantifiers and antonyms. The entailment decision is based on applying rewards and penalties over the semantic-similarity and shallow scores. Delmonte *et al.* [7] participated in the RTE-2 Challenge with an enhanced version of their previous system. The new system consists in new features based on heuristics such as: Augmented Head Dependency Structures, grammatical relations, negations and modal verbs. The motivation behind the approaches that use semantic features is that a pair with a strong similarity score holds an entailment relation.

III. PROPOSED METHOD

The RTE can be seen as a binary classification task where the entailment relations are the classes, and the RTE benchmark datasets are used to train a classifier [8].

Our RTE system is based on a supervised Machine Learning algorithm. We train the Machine Learning algorithm with similarity scores computed over the T-H pairs extracted from different classes of metrics .

1) *Features*: The features are based on scores computed by a similarity metric. The features are as follows:

Lexical Metrics: We use Word overlap (Equation 1), Cosine (Equation 2), Dice (Equation 3), Jaccard (Equation 4) and Overlap (Equation 5) metrics with a representation of Bag-of-Words (BoW) of the T-H pairs. For each metric we use as input different representations of the T-H pairs, such as tokens, lemmas and Part-of-Speech.

$$word_overlap(T, H) = |T \cap H| \quad (1)$$

$$cosine(T, H) = \frac{|T \cap H|}{\sqrt{|T| \times |H|}} \quad (2)$$

$$dice(T, H) = 2 \cdot \frac{|T \cap H|}{|T| + |H|} \quad (3)$$

$$jaccard(T, H) = \frac{|T \cap H|}{|T \cup H|} \quad (4)$$

$$opverlap(T, H) = \frac{|T \cap H|}{\min(|T|, |H|)} \quad (5)$$

Semantic Edit Distance: The TINE [9] is an automatic metric based on the use of shallow semantics to align predicates and their respective arguments between a pair of sentences. The metric combines a lexical matching with a shallow semantic component to address adequacy for Machine Translation evaluation. The goal of this metric is to provide a flexible way of align shallow semantic representations (semantic role labels) by using both the semantic structure of the sentence and the content of the semantic components.

Our first approach towards a Semantic Edit Distance metric consist in modify the TINE metric. The modified version of TINE is divided into two stages: i) The automatic alignment of predicates and ii) The Edition of the arguments between the aligned predicates.

In the alignment stage a set of Verbs between the T-H pair are aligned using VerbNet [10] and VerbOcean [11]. A verb in the Hypothesis is aligned to a verb in the Text if they have the same lemma or they are related according to the following heuristics: (i) the share at least one class in VerbNet; or (ii) the pair of verbs holds a relation in VerbOcean. For example, in VerbNet the verbs *spook* and *terrify* share the same class *amuse-31.1*, and in VerbOcean the verb *dress* is related to the verb *wear*. The output of this stage is a set of aligned verbs.

In the Edition stage we define three operations over arguments:

- 1) Deletion of an argument
- 2) Insertion of an argument
- 3) Substitution of an argument

The Deletion operation is applied if one of the arguments in H is missing in T, the Insertion operation is applied if one of the arguments in T is missing in H, and the Substitution operation is applied if the arguments in the T-H pair are of the same type but they have a different word realization. For each pair of verbs from the set supplied by the previous stage the Equation 6 is computed, and then the final score is the score average over the total of verbs Equation 7.

$$edition_score(T_v, H_v) = \frac{1}{number_of_operations}, \quad (6)$$

where T_v is an aligned verb in the Text, H_v is the corresponding aligned verb in the Hypothesis, and *number_of_operations* is the addition of the applied operations for each argument on T. In this version the metric does not give an importance weight for each operation.

$$semantic_score(T, H) = \frac{1}{n} \sum_{v=1}^n edition_score(T_v, H_v), \quad (7)$$

where n is the number of verbs. The following example shows how the computation of $edition_score(T_v, H_v)$ is performed:

T: *Recent Dakosaurus research comes from a complete skull found in Argentina in 0, studied by Diego Pol of Ohio State University, Zulma Gasparini of Argentinas National University of La Plata, and their colleagues.*

H: *A complete Dakosaurus was discovered by Diego Pol.*

- 1) extract verbs from T: $T_v = \{\text{comes, found, studied}\}$
- 2) extract verbs from H: $H_v = \{\text{discovered}\}$

- 3) similar verbs aligned with VerbNet (shared class discover-84-1-1): $V = \{\text{(study, discover)}\}$
- 4) Apply operations over T arguments:
 operation 1:insert A1 = {A complete Dakosaurus}
 operation 2:substitution A0 = {by Diego Pol}
 $edition_score(study, discover) = 1/2 = 0.5$

However, the alignment stage may not be able to match any verb. We use a back-off score metric in case the score of the Edit Distance is zero. The back-off score is based on shallow syntactic annotation or Chunking.

Chunking is a partial parser representation of text. It is an alternative to full parsing because it is more efficient and it is more robust. Chunks are non overlapping regions of text, and they are sequences of constituents which form a group with a grammatical role (e.g. NP noun group).

The chunking score is defined as the average of the number of similar chunks (in the same order) between the T-H pairs.

$$chunking(X, Y) = \frac{1}{m} \sum_{n=1}^m simChunk(x_n, y_n), \quad (8)$$

where m is the number of chunks in X , x_n is the n chunk tag and content in the same order, and $simChunk(x_n, y_n) = 1$ if the content and annotation of the chunk is the same, and $simChunk(x_n, y_n) = 0.5$ if the content of the chunk is different but the chunk tag is still the same.

The following example shows how the chunking $simChunk(x_n, y_n)$ works:

T: *Along with chipmaker Intel, the companies include Sony Corp., Microsoft Corp., NNP Co., IBM Corp., Gateway Inc. and Nokia Corp.*

H: *Along with chip maker Intel, the companies include Sony, Microsoft, NNP, International Business Machines, Gateway, Nokia and others.*

First, for each chunk the metric compares and scores the content of the tag if it is the same chunk group and if it is the same order of chunks.

Finally, the metric (Equation 8) computes the individual scores $chunking(T, H) = 0.64$.

With the previous metrics we build a vector of similarity scores used as features to train a Machine Learning algorithm. We use the development datasets from the RTE 1 to 3 benchmark to train SVM implementation from WEKA¹ without any parameter optimization.

In order to test our RTE system we divide the vector of features into different subsystems: (i) Baseline system which consist of lexical metrics scores, (ii) Semantic system which consist of the Edit Distance metric score, and (iii) Combination system which consist of a subset of the lexical features and the Edit Distance metric. The motivation to

¹<http://www.cs.waikato.ac.nz/ml/weka/>

Table I
EXAMPLE OF $simChunk(x_n, y_n)$ BACK-OFF SCORE

Tag	Content	Tag	Content	Score
PP	Along	PP	Along	1
PP	with	PP	with	1
NP	chipmaker Intel	NP	chip maker Intel	0.5
NP	the companies	NP	the companies	1
VP	include	VP	include	1
NP	Sony Corp.	NP	Sony	0.5
NP	Microsoft Corp.	NP	Microsoft	0.5
NP	IBM Corp.	NP	International Business Machines	0.5
NP	Gateway Inc.	NP	Gateway	0.5
NP	Nokia Corp.	NP	Nokia and others.	0.5

divide the vector of features is that the semantic feature is not enough to predict entailment, and a semantic feature will help to improve the performance of a simple lexical system. Moreover, we use a feature selection algorithm to select the best combination of lexical features to improve the Baseline system. Then, these subset of lexical features is used for the Combination system, and the feature selection step results in a new Baseline system.

IV. EXPERIMENTAL RESULTS

We compare our method with ML-based methods, and with methods that use a SRL representation as one of its features or edit distance algorithms. We use the RTE-1, RTE-2, and RTE-3 development datasets to train the classifiers. The data used for classification are the test datasets of the RTE Challenge. The experimental results for the both Baseline systems are summarized in Table II. The Baseline system uses the following features:

- 1) Tokens with Word overlap, Cosine, Dice, Jaccard, and Overlap
- 2) Lemmas with Word Overlap, Cosine, Dice, Jaccard, and Overlap
- 3) PoS with Word overlap, Cosine, Dice, Jaccard, and Overlap

The result is a vector of 15 features where each representation of the T-H pair is scored by a different lexical metric. For example, with the previous 15 features over the RTE-3 development data set. The result for using a SVM and a 10-fold cross-validation is an Accuracy of 63.25%. Then, we use an in-house feature selection algorithm (genetic algorithm), in order to extract the best combination of the previous features. The design of the genetic algorithm is as follows:

- The chromosome of the individual is composed of three genes. Each gene is one of the representations (i.e. token, lemma and PoS), and each representation could be measure with one of the similarity metric (i.e. word overlap, cosine, dice...)
- The fitness function is the accuracy of the features over a given development set
- A population of 80 individuals

- A crossover probability of 90%
- A mutation probability of 10%
- The selection method is the Roulette scheme
- The crossover strategy is one point
- The offspring comes from two parents
- The tree best individual of the previous generation is preserved

The features chosen by the feature selection algorithm are:

- 1) Tokens with Overlap
- 2) Lemmas with Cosine
- 3) Lemmas with Overlap

The result for the Baseline system over the RTE-3 development dataset with a 10-fold-cross-validation is and Accuracy of 63.8%. The results of the both Baseline systems are similar in the 10-fold-cross-validation. In addition, we compute the McNemar’s test over the Baseline systems, and they are not statistical-significant different. The Baseline with feature selection shows worst results over the test datasets.

Table II
ACCURACY RESULTS FOR THE BASELINE SYSTEM AND THE BASELINE SYSTEM WITH A SUBSET OF THE LEXICAL METRICS OVER THE TEST DATASETS

Method	RTE-1	RTE-2	RTE-3
Baseline	51.38%	58.5%	59.5%
Baseline with feature selection	50.75%	58.37%	59%

For the Edit distance system we use the same configuration for the ML as the Baseline. The experimental results for the Semantic Edit Distance system are summarized in Table III

Table III
ACCURACY RESULTS FOR SEMANTIC EDIT DISTANCE METRIC OVER THE TEST DATASETS

Method	RTE-1	RTE-2	RTE-3
Semantic Edit Distance	50.25%	51.87%	51.25%

Table IV shows the overall accuracy results of the RTE test datasets against our method. Our method is close to

Table IV
COMPARISON WITH OVERALL ACCURACY RESULTS OVER THE RTE
TEST DATASETS

Challenge	Our method	Average	Best
RTE-1	51.5%	55.12%	70.00%
RTE-2	57.87%	58.62%	75.38%
RTE-3	59.37%	61.14%	80.00%

the average performance but far from the best method. However, the related work are complex systems. In contrast, our method relies on simple features.

Table V shows the comparison with previous works, our system is comparable with Kouylekov and Magnini [2]. However, the new metric improves in a small degree the proposed baseline with feature selection. We discuss with a few examples some of the common errors made by the the alignment step of the semantic edit distance metric. Overall, we consider the following categories of errors:

- 1) Lack of coverage of the ontologies.

T: *This year, women were awarded the Nobel Prize in all fields except physics.*

H: *This year the women received the Nobel prizes in all categories less physical.*

The lack of coverage in the VerbNet ontology prevented the detection of the similarity between *receive* and *award*.

- 2) Matching of unrelated verbs.

T: *If snow falls on the slopes this week, Christmas will sell out too, says Schiefert.*

H: *If the roads remain snowfall during the week, the dates of Christmas will dry up, said Schiefert.*

In VerbOcean *remain* and *say* are incorrectly said to be related. VerbOcean was created by a semi-automatic extraction algorithm [11] with an average accuracy of 65.5%.

- 3) Incorrect tagging of the semantic roles by the semantic parser SENNA².

T: *Colder weather is forecast for Thursday, so if anything falls, it should be snow.*

H: *On Thursday, must fall temperatures and, if there is rain, in the mountains should.*

The position of the predicates affects the SRL tagging. The predicate *fall* has the following roles (A1, V, and S-A1) in the reference, and the following roles (AM-ADV, A0, AM-MOD, and AM-DIS) in the hypothesis. As a consequence, the metric cannot attempt to match the fillers. Also, SRL systems do not detect phrasal verbs, where the action *putting people off* is similar to *discourages*.

²SENNA, <http://ml.nec-labs.com/senna/>

Then, we analyze the datasets in order to discuss the impact of the new metric over the Combination system, where the quality of the semantic parser and the coverage of the ontologies can be reasons that affects the performance of this method. For example, in the RTE-1 test dataset with 800 T-H pairs the coverage of the new metric is 491 pairs. Which means that the metric only scores that amount of pairs. In the RTE-3 dataset, which is the model with the best result, with 800 T-H pairs, but the coverage increases to 556 pairs. Thus, the method reduces the amount of errors with additional semantic-scored pairs. In addition, the remaining pairs out of the coverage of the semantic score are scored by the back-off, and this back-off shows to be a poor predictor for entailment.

V. CONCLUSIONS

We have presented a ML-based system for RTE based on a new similarity metric. We proposed a simple baseline based on similarity metrics over different representations of text, as well as a method to select the best combination of the previous metrics with the representations. We also proposed a preliminary approach towards a semantic edit distance metric based on predicate-argument representation of the T-H pairs, and the use of a back-off score based on shallow parsing. While the baseline method has the best performance, the addition of the semantic information improves the performance, also to a small degree. The evaluation of the datasets shows that the coverage of the semantic metric affects the overall performance of the system, and the back-off score is a poor predictor for entailment.

For future work we plan to improve the semantic edit distance by automatically learn the weights for each type of operation, and use a different way to compute the set of applied operations. A direction to compute the set of applied operations can be to measure the difference between the language models perplexity of the modified T against H. The motivation to use language models is that with each applied operation over T the difference between the language models will drop.

In addition, we plan to change the back-off score by a better syntactic scoring or even combine our semantic metric with a different semantic approach to RTE [15]. Information about syntactic valencies and predicate-argument structure extracted from machine-readable dictionaries [16], [17] can also prove to be useful for this task.

ACKNOWLEDGMENTS

The work was partially supported by the Mexican Government, specifically, by the Mexican National Council for Science and Technology (CONACYT) via scholarship reference 309261 to the first author and CONACYT 50206-H and SIP-IPN 20121823 projects; by Governments of India and Mexico under the CONACYT-DST India (CONACYT

Table V
COMPARISON WITH PREVIOUS WORKS OVER THE RTE TEST DATASETS

Method	RTE-1 (Accuracy)	RTE-2 (Accuracy)	RTE-3 (Accuracy)
Roth and Sammons [4]	-	-	65.56%
Burchardt and Frank [12], Burchardt <i>et al.</i> [5] Delmonte <i>et al.</i> [6], Delmonte <i>et al.</i> [7], Delmonte <i>et al.</i> [13]	54.6%	59.8%	62.62%
Kouylekov and Magnini [2], Kouylekov and Magnini [14]	55%	60%	-
Combination system	51.5%	57.87%	59.37%

122030) project “Answer Validation through Textual Entailment”, and the Seventh Framework Programme of European Union, project 269180 “Web Information Quality Evaluation Initiative (WIQ-EI)”.

REFERENCES

- [1] I. Dagan and O. Glickman, “The pascal recognising textual entailment challenge,” in *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.
- [2] M. Kouylekov and B. Magnini, “Recognizing textual entailment with tree edit distance algorithms,” in *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.
- [3] E. Cabrio, M. Kouylekov, and B. Magnini, “Combining specialized entailment engines for rte-4,” in *Proceedings of the Text Analysis Conference 2008*, Gaithersburg, Md., 17–19 November 2008, 2008.
- [4] D. Roth and M. Sammons, “Semantic and logical inference model for textual entailment,” in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague: Association for Computational Linguistics, June 2007, pp. 107–112.
- [5] A. Burchardt, N. Reiter, S. Thater, and A. Frank, “A semantic approach to textual entailment: System evaluation and task analysis,” in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague: Association for Computational Linguistics, June 2007, pp. 10–15.
- [6] R. Delmonte, S. Tonelli, M. A. Piccolino Boniforti, A. Bristot, and E. Pianta, “Venses - a linguistically-based system for semantic evaluation,” in *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.
- [7] R. Delmonte, A. Bristot, M. A. P. Boniforti, and S. Tonelli, “Coping with semantic uncertainty with venses,” in *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy, 2006.
- [8] I. Dagan, B. Dolan, B. Magnini, and D. Roth, “Recognizing textual entailment: Rational, evaluation and approaches - erratum,” *Natural Language Engineering*, vol. 16, no. 1, p. 105, 2010.
- [9] M. Rios, W. Aziz, and L. Specia, “Tine: A metric to assess mt adequacy,” in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 116–122.
- [10] K. K. Schuler, “Verbnet: A broad-coverage, comprehensive verb lexicon,” Ph.D. dissertation, University of Pennsylvania, 2006.
- [11] T. Chklovski and P. Pantel, “VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations,” in *Proceedings of EMNLP 2004*, D. Lin and D. Wu, Eds., Barcelona, Spain, Jul. 2004, pp. 33–40.
- [12] A. Burchardt and A. Frank, “Approaching textual entailment with lfg and framenet frames,” in *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy, 2006.
- [13] R. Delmonte, A. Bristot, M. A. Piccolino Boniforti, and S. Tonelli, “Entailment and anaphora resolution in rte3,” in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague: Association for Computational Linguistics, June 2007, pp. 48–53.
- [14] M. Kouylekov and B. Magnini, “Tree edit distance for recognizing textual entailment: Estimating the cost of insertion,” in *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy, 2006.
- [15] P. Pakray, U. Barman, S. Bandyopadhyay, and A. Gelbukh, “A statistics-based semantic textual entailment system,” in *Proceedings of the 10th Mexican international conference on Advances in Artificial Intelligence - Volume Part I*, ser. MICAI’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 267–276.
- [16] N. A. Castro-Sánchez and G. Sidorov, “Analysis of definitions of verbs in an explanatory dictionary for automatic extraction of actants based on detection of patterns,” *Lecture Notes in Computer Science*, vol. 6177, pp. 233–239, 2010.
- [17] —, “Automatic acquisition of synonyms of verbs from an explanatory dictionary using hyponym and hyperonym relations,” *Lecture Notes in Computer Science*, vol. 6718, pp. 322–331, 2011.