Textual Entailment beyond Semantic Similarity Information

Sonia Vázquez[1], Zornitsa Kozareva[1], and Andrés Montoyo[1]

Department of Software and Computing Systems
University of Alicante
{svazquez,zkozareva,montoyo}@dlsi.ua.es

Abstract. The variability of semantic expression is a special characteristic of natural language. This variability is challenging for many natural language processing applications that try to infer the same meaning from different text variants. In order to treat this problem a generic task has been proposed: Textual Entailment Recognition. In this paper, we present a new Textual Entailment approach based on Latent Semantic Indexing (LSI) and the cosine measure. This proposed approach extracts semantic knowledge from different corpora and resources. Our main purpose is to study how the acquired information can be combined with an already developed and tested Machine Learning Entailment system (MLEnt). The experiments show that the combination of MLEnt, LSI and cosine measure improves the results of the initial approach.

1 Introduction

In our daily life, we use different expressions to transmit the same meaning. Therefore, Natural Language Processing (NLP) applications, such as Question Answering, Information Extraction, Information Retrieval, Document Summarization, Machine Translation among others, need to identify correctly the sentences that have different surface forms, but express the same meaning. This semantic variability task is very important and its resolution leads to improvement in system's performance [1].

The task of Textual Entailment (TE)[2][3] consists in given two text fragments, set whether the meaning of one text (the hypothesis) can be inferred from the meaning of the other text. For example: Text(He died of blood loss) and Hypothesis(He died bleeding), in this case, the hypothesis infer the same meaning than test. For the resolution of the TE task, different approaches have been developed [4] [5] [6] [7].

In this paper, we describe a novel approach for the modelling and extraction of semantic information with Latent Semantic Indexing (LSI)[8] and the cosine measure. In LSI, the traditional approaches use large corpora to represent a term-document matrix through which the semantic information is obtained. However, we propose an approach where the corpora consist of the TE text/hypothesis sentences and the vector space is a text-hypothesis/hypothesis-text matrix. In addition, LSI is applied with the WordNet Domains resource [9]. Instead of measuring the similarity of a term-document matrix, we construct a term-domain matrix. To our knowledge, current researchers did not take advantage of such information. Moreover, we have used the cosine measure with two types of information: from corpus and from Relevant Domains (RD) resource [10].

In order to show the contribution of our approach, we conduct an exhaustive evaluation. In the experiments, we study and compare the traditional corpus-based approach to the ones we propose. Then, we examine the effect of the incorporation of the new semantic similarity information to our previous TE system [11].

2 Semantic knowledge representation with LSI

LSI is a computational model that takes advantage of a property of natural language: words of the same semantic field usually appear in the same context. This model establishes word's relations from a large linguistic corpus using a vectorial-semantic space where all terms are represented (term-document matrix). In order to obtain appropriate information, terms have to be distributed in documents, paragraphs or sentences. This distribution will determine which is the co-occurrence among different terms and the threshold of using other terms in the same context. In other words, LSI extracts relations among terms and documents and tries to reduce the current noise in these relations. For this purpose, and once the term-document matrix is obtained, LSI uses a variant of factorial analysis: the Singular Value Decomposition (SVD). This technique uses a recursive algorithm to decompose the term-document matrix into three other matrices, that contain singular vectors and singular values. These matrices show a breakdown of the original data into linearly independent factors. Moreover, a great number of these factors are very small and can be ignored in order to obtain an approximated model with less factors. The final result is a reduced model of the initial term-document matrix that will be used in order to establish word similarities.

This section presents how we extract information from different linguistic resources and how we apply LSI method in order to extract word similarity information from a semantic space. We are interested in how LSI contributes to resolve TE because it has been not applied before in such task.

For our purpose, we have done different experiments using three types of corpus. The first one is British National Corpus (BNC)[12] and we build the LSI semantic space using the documents in BNC. The obtained term-document matrix will be used by the LSI method. The second corpus is obtained from sentences of Text and Hypothesis of RTE2. In this case, we build two type of matrices: text-hypothesis matrix (rows are the text sentences and columns are the hypothesis sentences) and hypothesis-text matrix (rows are hypothesis sentences and columns are the text sentences). The main purpose of using these different type of matrices is to measure how important the words are in the semantic space in order to extract relevant information. And the last type of corpus is obtained from WordNet Domains resource. In this case we build a term-domain matrix using the information provided by WordNet Domains [9].

All these matrices representation types are described in subsections below.

2.1 Semantic space from corpus

British National Corpus has a collection of about 4000 documents obtained from national newspapers, specialist periodicals and journals for all ages and interests... This corpus provides useful information in order to establish word relations from their frequency in documents.

For our purpose, we build a term-document matrix where rows represent all the possible terms in the corpus and columns represent all the documents. In our first approximation, we extract all words previously stemmed and compute how many times each word appears in each document. This information is provided to the LSI module in order to obtain a new conceptual space.

Once we have our conceptual space based on the information provided by the BNC corpus we can establish how similar each pair of sentences are. In our case, we want to infer if each T_i-H_i pair (where $i = 1..n$ is the number of pairs in RTE2) infer the same meaning. For this purpose and using LSI method, we have done two types of experiments: one extracting the 20 more relevant documents to each sentence and other extracting the 800 more relevant words of each sentence. The final result for each type of experiment is a normalized value between 0-1 to illustrate how similar are each pair of sentences (the more closer to 1 the more similar they are).

2.2 Semantic space from Text-Hypothesis

In this section we present another kind of experiments using as source of information the words of Text and Hypothesis sentences. In this case, we want to establish how similar are each T-H pair by using as semantic space: Text sentences or Hypothesis sentences. In other words, we build two different types of matrices: one using Text sentences as corpus and other using Hypothesis sentences as corpus.

The Hypothesis-Text matrix has one column for each Text sentence. So, in order to establish how similar H-T are, we compute for each Hypothesis sentence which are the most relevant Text sentences according our conceptual space. The result is a list of the 20 most relevant Text sentences with a similarity value associated. If the Text couple of the Hypothesis we are searching is among the 20 Text sentences extracted, we extract the similarity value given by the LSI method (between 0-1) in other case, H-T has a value of 0 similarity.

The same procedure is used to build the Text-Hypothesis matrix and compute how similar T-H pairs are.

2.3 Semantic space from WordNet Domains

In this section we show how to create a term-domain matrix from the information of WordNet Domains in order to obtain a new semantic space. So, the first step is obtaining the set of domain labels because we will have one column for each domain label in our term-domain matrix. In this case we have a hierarchy with about 200 labels.

Once we know how many domain labels are, we need to extract which words are related to each domain label. In other words, we must obtain a list of words for each domain label using the information of WordNet Domains. In this step we extract the information of WordNet glosses and assign to each word its associated domain. That is to say, we have grouped all word senses (with their gloss and examples) in domain labels in order to do pairs of word-domain. An example of how domain labels are assigned to each word sense is showed in Table 1.

Synset	Domain	Word Sense	Gloss
02330776	music	Brass#1	a wind instrument that consists of a brass tube (usually of variable length) blown by means of a cup-shaped or funnel-shaped mouthpiece
06071657	administration politics	Brass#2	the persons (or committees or departments etc.) who make up a governing body and who administer something; ”he claims that the present administration is corrupt”;
03792240	factotum	Brass#3	impudent aggressiveness; ”I couldn’t believe her boldness”; ”he had the the effrontery to question my honesty”
02331254	factotum	Brass#4	an ornament or utensil made of brass
02331144	tourism	Brass#5	a memorial tablet made of brass

Table 1. Labelling brass senses with domains

In this case, the word brass has associated for each sense different domain labels and in WordNet Domains all word senses in are tagged in the same way as brass in Table 1. To obtain a list of word-domain pairs, we assume that words of glosses are semantically closer to the word sense defined. So, we assign for each word of the gloss the same domains assigned for the word sense defined. For example, brass#1 has the gloss *”a wind instrument that consists of a brass tube (usually of variable length) blown by means of a cup-shaped or funnel-shaped mouthpiece”*, and the domain *”music”*, therefore, each word in its gloss is related to domain *”music”*[1].

Once we obtain all word-domain pairs, we build a word-domain matrix in order to obtain a new semantic space with LSI, based on information from domain classification.

3 Application of the cosine measure

In order to identify different similarities among words we need to establish one way to measure the degree of similarity. In this work we choose a vector based approach, the cosine measure. This approach measures the distance between two words using co-occurrence vectors. Each word is represented with one co-occurrence vector and the degree of similarity between two words is obtained by measuring the distance between its associated co-occurrence vectors. So, in order to obtain co-occurrence vectors there are different types of lexical relationships that can be used. The traditional corpus-based approach is based on building a type of vectors named word co-occurrence vectors. This type of vectors represent a word by their patterns with other words in a corpus. In other words, we can measure the similarity between two words using grammatical relations (co-occurrence of words in specific syntactic relations) or non grammatical relations (co-occurrence of words in a n-words window). However, we can consider other type of co-occurrence vectors using an alternative representation: document co-occurrence

[1] In order to obtain word-domain pairs we only consider nouns, verbs, adjectives and adverbs

vectors. In this case, relations among words are extracted from a set of documents and the similarity between each pair of words is computed by measuring their overlap in the set of documents.

Next subsections present a description of how we obtain co-occurrence vectors to measure the similarity between both sentences Text and Hypothesis. We introduce two different types of co-occurrence vectors, one based on corpus information and the other based on a lexical resource (Relevant Domains) obtained from a lexical database.

3.1 Document frequency

In our first approach we study the effect of using the cosine measure with the information provided by the BNC. This corpus provides a set of about 4000 documents and we establish the similarity between two words using document co-occurrence vectors. Our purpose is to establish how similar are the Text (T) and Hypothesis (H) sentences by their semantic distance.

The first step is representing T and H with vectors. Each vector has around 4000 attributes, one for each document in BNC. In our case, each vector represents one sentence (T or H). So, in order to give value to each attribute of the vector we need to compute which is the frequency of all words in the sentence in each documents in the corpus. Notice that the number of words in T and H is different, so, we need to normalize the results obtained according to the number of words of each sentence.

In addition, once obtained the information provided by frequencies of words we calculate the Inverse Document Frequency (*idf*). This measure is commonly used in Information Retrieval and provides high values for rare words and low values for common words. The Equation 1, gives the *idf* formula where N is the total number of documents and n_w is the number of documents that contains the word w. The *idf* is the final value used for representing each attribute of the co-occurrence vectors. This value is referred as Document Frequency (DF).

$$idf_w = \log\left(\frac{N}{n_w}\right) \tag{1}$$

Once obtained the document co-occurrence vectors, we can measure the similarity between two sentences (T and H) by the value of the cosine (Equation 2).

$$\cos(T,H) = \frac{T \cdot H}{|T|\,|H|} = \frac{\sum_{i=1}^{n} T_i \cdot H_i}{\sqrt{\sum_{i=1}^{n} {T_i}^2} \cdot \sqrt{\sum_{i=1}^{n} {H_i}^2}} \tag{2}$$

In our study, we use different cosine boundaries in order to establish which are the most appropriate values to extract correct inferences between T and H. The results of applying this measure are showed in Table 3.

3.2 Relevant Domains

This section presents a second approach to infer semantic relations between T and H. In this approach, we obtain the cosine measure using Relevant Domains (RD) [10] information in order to represent domain co-occurrence vectors. The RD resource is obtained from WordNet Domains (WND) [9] that is an extension of WordNet. The characteristics and structure of this resource has been explained in section 2.3.

In order to extract the RD we use the words of WordNet glosses. In Table 1 we have an example of how the glosses are related to domains. We label the words of glosses with their appropriate domain. This information is used to compute how relevant is one word to each domain. So, once we know how many times one word appears with one domain in the whole lexical database, we can establish with the Association Ratio (AR) [13] (Equation 3), which are the most relevant domains to this word.

$$AR(w,D) = Pr(w|D)\, log_2 \frac{Pr(w|D)}{Pr(w)} \tag{3}$$

Therefore, the RD contains all words of WordNet Domains with their most relevant domains sorted by the AR. For example, here we have an extraction of how AR is calculated for the word *"organ"*: *Organ*{*Surgery-0.189502, Radiology-0.109413, Sexuality-0.048288, Optics-0.048277, Anatomy-0.047832, Physiology-0.029388, ...- ...*}.

In our work, we use this resource in order to build domain co-occurrence vectors. This kind of vectors have as many attributes as many domains. To measure the distance between T and H sentences we extract all words of each sentence, obtain which are their RD and build the domain co-occurrence vectors. Therefore, once we have each pair of domain co-occurrence vectors we calculate their semantic distance with the cosine measure.

One of the reasons of using this type of resource is because we want to establish how corpus dependent is the cosine measure. In other words, we can extract word frequencies from different corpora and obtain different results when we calculate the distance between the same pair of sentences. So, with the RD resource we try to avoid the corpus dependence because word-domain pairs are extracted from a lexical database and relations are obtained according their meanings and not according a specific field or document classification.

4 Experimental results

This section presents a set of experiments using our different approaches. In one hand, we have done experiments with LSI using as corpus the BNC, WordNet Domains, Text sentences and Hypothesis sentences. And in the other hand, we have done experiments with the cosine measure using a document frequency approach and a new approach using Relevant Domains resource. Moreover, we have combined these different approaches with our machine-learning system in order to study the effect of adding this new information. As a result, we find that the combination of LSI and cosine with the machine-learning system improves the textual entailment results.

4.1 The *RTE2* data

For our experimental setup, we use the development and test data sets provided by the Second Recognizing Textual Entailment Challenge (RTE2)[2]. The examples in these

[2] http://www.pascal-network.org/Challenges/RTE2/

data sets have been extracted from real Information Extraction (IE), Information Retrieval (IR), Question Answering (QA) and Text Summarization (SUM) applications. The corpus includes 1600 English text-hypothesis entailment examples, of which 800 are used as a development data and the remaining 800 pairs as a test data. In RTE2 the corpus is balanced, 50% are true examples and the other 50% are false examples.

The performances of our experiments are determined with the RTE2 evaluation script[3]. According to the script, systems are ranked and compared by their accuracy scores.

4.2 LSI

The LSI experiments show the influence of the usage of different corpora through which the sense of two sentences can be inferred.

As we describe in Section 2 we build the LSI initial matrix from different types of corpora. Therefore, for each kind of corpus we have obtained different results for the TE task. The information used in each experiment is explained here [4]:

- **BNC corpus** *(LSI_BNC_NoTag)*. Results using lemmatized words from BNC.
- **H sentences** *(LSI_LemaH, LSI_NoLemaH)*. Results using as corpus H sentences and building two different matrices: one with lemmatized words and another with no lemmatized words.
- **T sentences***(LSI_LemaT, LSI_NoLemaT)*.Results using as corpus T sentences and building two different matrices: one with lemmatized words and another with no lemmatized words.
- **Relevant Domains** *(LSI_RD)*. Results using the relevant domains of each T_sentence and each H_sentence.

Table 2 shows the results obtained from different experiments with LSI.

As we can see, the best results are obtained using the Text sentences (LSI _LemaT) and the Relevant Domains (LSI_RD). The first approach uses as corpus all Text sentences, and the Hypothesis sentences are used as input to the LSI module. In this case, the results are 56.87% for the development data set and 54.25% for the test data set. This results are better than the (LSI_LemaH) because Text sentences provide more lexical information that can be used. So, in order to infer whether 2 sentences have the same meaning we need an appropriate base context. The second approach uses as corpus the RD resource. In this case, the initial matrix is obtained from the information of WordNet Domains. We use this semantic space in order to extract how similar are T-H sentences. As a result, we obtain a percentage of 56.98% for the development data set and 54.51% to the test data set. In this case, the results are good because words are semantically related according to their associated domains and this information improves the results of QA and SUM.

The other experiments reveal that we have not enough information to establish a correct TE detection, so we can use this information as a random baseline.

[3] http://www.pascal-network.org/Challenges/RTE2/Evaluation/

[4] Each experiment is preceded by *dev* (development dataset) or by *test* (test dataset)

Sets	Acc.	IE	IR	QA	SUM
devLSI_BNC_NoTag	49.90	49.87	49.15.	50.15	50.43
devLSI_LemaH	53.25	52.00	48.00	54.00	59.00
devLSI_NoLemaH	50.17	50.15	50.03	50.22	50.28
devLSI_LemaT	**56.87**	**51.50**	**58.00**	**56.50**	**61.50**
devLSI_NoLemaT	52.88	50.50	53.00	48.00	60.00
devLSI_RD	**56.98**	**52.25**	**58.60**	**56.83**	**60.25**
testLSI_BNC_NoTag	49.67	49.43	49.00	50.02	50.24
testLSI_LemaH	49.38	52.50	48.50	49.00	47.50
testLSI_NoLemaH	53.37	50.50	54.00	49.00	60.00
testLSI_LemaT	**54.25**	**50.50**	**48.00**	**57.00**	**61.50**
testLSI_NoLemaT	53.63	52.50	50.00	50.00	62.00
testLSI_RD	**54.51**	**50.55**	**48.53**	**56.73**	**62.25**

Table 2. *Results for the LSI*

4.3 Cosine

This experimental section shows the result of the measurements of the similarity of the sentences with the cosine measure. In Table 3, we present the results of the traditional document frequency cosine approach and those of the RD approach. The document frequency reaches 52% accuracy and can be used as a TE baseline. Both the development and test data sets reached 54% with the RD experiment. This similarity of performance is due to the fact that the context information given by the sentences is not very representative and does not provide enough knowledge. Therefore, on its own the cosine measure cannot establish the TE relation between the sentences, but still can be useful when combined with other information sources.

Sets	Acc.	IE	IR	QA	SUM
devCosine_DF	52.60	48.63	47.32	55.13	59.32
devCosine_RD	54.25	50.50	48.00	57.00	61.50
testCosine_DF	52.18	46.13	49.43	55.34	57.83
testCosine_RD	54.00	46.50	56.50	56.00	57.00

Table 3. *Results for the cosine measure*

4.4 Combination of MLEnt with LSI and the cosine measure

The experiments revealed that LSI and the cosine are not powerful enough to establish the correct TE relation of two sentences. However, they still contribute and provide useful information. We believe that when these techniques are combined with other knowledge sources, the TE inference can be improved. We used a previous Machine learning TE system (MLEnt) and added the information provided for both LSI and cosine measure. Our purpose is studying whether the combination of MLEnt with semantic information improves the previous results.

In Table 4 there are several experiments combining the MLEnt system with both LSI and cosine measures. We can distinguish two types of experiments: one with the previous MLEnt system and other with the combination of LSI and cosine measure. Each experiment is detailed next:

– **MLEnt with previous features** *(MLEnt_Lex, MLEnt_Sem)*. Results of the previous MLEnt system with Lexical or Semantical features.
– **MLEnt with LSI** *(MLEnt_Lex_LSI_LemaT, MLEnt_Sem_LSI_LemaT)*. Results of the previous MLEnt system with LSI. In this case, we use as corpus for LSI the Text sentences with lemmatized words.
– **MLEnt with cosine** *(MLEnt_Lex_cosine,MLEnt_Sem_cosine)*. Results of the previous MLEnt system with the cosine measure. In this case, cosine is obtained from Relevant Domains.
– **MLEnt with LSI and cosine** *(MLEnt_Lex_LSI_LemaT_cosine, MLEnt_Sem_LSI_LemaT_cosine)*. Results of the previous MLEnt system with LSI and cosine measure. In this case, we use LSI with T-sentences and cosine with Relevant Domains.

In order to measure the effect of adding semantic information in the MLEnt system we have selected the best results obtained in the experiments with LSI and cosine.

Sets	**Acc.**	**IE**	**IR**	**QA**	**SUM**
devMLEnt_Lex	56.87	49.50	55.50	51.00	71.50
devMLEnt_Sem	60.12	54.00	61.00	59.00	66.50
devMLEnt_Lex_LSI_LemaT	**62.03**	**56.13**	**62.53**	**60.32**	**69.15**
devMLEnt_Lex_cosine	56.91	49.45	55.62	52.13	70.43
devMLEnt_Lex_LSI_LemaT_cosine	57.13	49.50	55.50	52.50	71.00
devMLEnt_Sem_LSI_LemaT	**62.56**	**57.13**	**62.83**	**60.54**	**69.75**
devMLEnt_Sem_cosine	60.21	54.13	61.06	59.14	66.54
devMLEnt_Sem_LSI_LemaT_cosine	**61.75**	**56.00**	**59.50**	**62.50**	**69.00**
testMLEnt_Lex	51.75	52.00	53.50	55.50	46.00
testMLEnt_Sem	54.25	50.00	55.50	47.50	64.00
testMLEnt_Lex_LSI_LemaT	**55.01**	**51.23**	**55.83**	**47.96**	**65.03**
testMLEnt_Lex_cosine	52.57	49.50	44.95	53.73	62.13
testMLEnt_Lex_LSI_LemaT_cosine	54.87	46.50	53.00	56.00	64.00
testMLEnt_Sem_LSI_LemaT	**56.18**	**52.03**	**56.53**	**50.14**	**66.03**
testMLEnt_Sem_cosine	54.42	50.22	55.62	47.61	64.25
testMLEnt_Sem_LSI_LemaT_cosine	**56.50**	**53.00**	**58.00**	**57.50**	**57.50**

Table 4. *Results for the combination of MLEnt with LSI and the cosine measure*

As Table 4 shows, the experiments carried out combining LSI and cosine information improve the previous results of the MLEnt system. We noticed that adding this information as a new feature to our MLEnt system obtain better results. So, the addition of semantic information is a good way to improve the results in a MLEnt system.

In fact, the best score is about 62% for the development data set and about 57% for the test data set. This score was obtained in an experiment that combined the results of LSI, cosine and MLEnt system.

In conclusion, we can assume that LSI and cosine measure provide useful information that can improve a previous a machine-learning entailment system.

5 Conclusions

This paper presents a TE approach based on semantic similarity information obtained by the LSI and the cosine measure. Initially, we compare the influence of different corpora such as the BNC and the text-hypothesis sentence space. The experiments show that the results of a big corpus do not influence so much the one produced by the text-hypothesis space. The information in a corpus depends on the domain or the topic and can influence the relevance of a word or a whole sentence. In order to avoid such dependencies, we propose two approaches: LSI and cosine measure based on Relevant domains resource. These approaches consider information from a static resource, WordNet Domains lexical data base, that is more relevant than a dynamic corpus where the frequency ratio or presence of words changes. Therefore, we noticed that the results using the development and test data are similar reaching 54% for the cosine and 54.5% for the LSI methods.

Once we studied the contribution of LSI and the cosine, we noticed that the provided information by these techniques is not sufficient for the correct and ample recognition of TE. For this reason, we conducted another experiment, where the combination of LSI, the cosine and an already existing machine-learning based TE system were studied. The exhaustive experiments show how this combination improves results with 61.75% for the development data set and 56.50% for the test data set.

In conclusion, we noticed that LSI is a powerful NLP tool which serves for the extraction of semantic information and can improve the results of an existing machine-learning system. We explored different functions of this technique, however in the future, we want to use its properties in order to extract synonym, antonym and other type of word relations. Moreover, in this work the semantic similarity is evaluated considering the whole sentence, which introduces lots of noise. Therefore, we plan to study the influence of using syntagmatic information instead of using the whole sentence.

Acknowledgements

This research has been partially funded by the Spanish Government under project CICyT number TIC2003-07158-C04-01 and PROFIT number FIT-340100-2004-14 and by the Valencia Government under project numbers GV04B-276.

References

1. Kozareva, Z., Montoyo, A.: The role and the resolution of textual entailment for natural language processing applications. In: 11th International Conference on Applications of Natural Language to Information Systems (NLDB). (2006)

2. Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment. (2005)
3. Dagan, I., Glickman., O.: Probabilistic textual entailment: Generic applied modeling of language variability. In: PASCAL Workshop on Learning Methods for Text Understanding and Mining. (2004)
4. Akhmatova, E.: Textual entailment resolution via atomic propositions. (In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.) 61–64
5. Herrera, J., Peñas, A., Verdejo., F.: Textual entailment recognition based on dependency analysis and wordnet. (In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment,2005.)
6. Jijkoun, V., de Rijke, M.: Recognizing textual entailment using lexical similarity. (In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment,2005.)
7. Montes, M., Gelbukh, A., López, A., Baeza-Yates, R.: Flexible comparison of conceptual graphs. In: DEXA. Lecture Notes in Computer Science, Springer-Verlag (2001) 102–111
8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic indexing. In: Journal of the American Society for Information Science. Volume 41. (1990) 321–407
9. Magnini, B., Cavaglia, G.: Integrating Subject Field Codes into WordNet. In Gavrilidou, M., Crayannis, G., Markantonatu, S., Piperidis, S., Stainhaouer, G., eds.: Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, Athens, Greece (2000) 1413–1418
10. Vázquez, S., Montoyo, A., Rigau, G.: Using relevant domains resource for word sense disambiguation. In: IC-AI. (2004) 784–789
11. Kozareva, Z., Montoyo, A.: Mlent: The machine learning entailment system of the university of alicante. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment. (2006)
12. Aston, G.: The british national corpus as a language learner resource. (In: TALC 96)
13. Church, K., Hanks, P.: Word association norms, mutual information and lexicograhy. Computational Lingüistics **16** (1990) 22–29