The DLSIUAES Team's Participation in the TAC 2008 Tracks

Alexandra Balahur, Elena Lloret, Óscar Ferrández, Andrés Montoyo,
Manuel Palomar and Rafael Muñoz
Universidad de Alicante,
Departamento de Lenguajes y Sistemas Informáticos
Apartado de Correos 99, 03080 Alicante
{abalahur, elloret, ofe, montoyo, mpalomar, rafael}@dlsi.ua.es

Abstract. In this paper we present the DLSIAUES team's participation in the TAC 2008 Opinion Pilot and Recognizing Textual Entailment tasks. Structured in two distinct parts corresponding to these tasks, the paper presents the opinion and textual entailment systems, their components, as well as the tools and methods used to implement the approaches taken. Moreover, we describe the difficulties encountered at different steps and the distinct solutions that were adopted. We present the results of the evaluations performed within TAC 2008, analyze them and comment upon their significance. Finally, we conclude on the performed experiments and present some of the lines for future work.

Overview

Three tracks were defined in this year's TAC edition (TAC 2008): Question Answering, Recognizing Textual Entailment and Summarization. The Summarization track included two tasks: the Update Summarization and the Opinion Summarization Pilot task. The DLSIUAES team participated in two of these tracks, the Recognizing Textual Entailment track and the Opinion Summarization Pilot task within the Summarization track.

This paper is mainly divided into two parts with respect to the tasks in which the DLSIUAES team participated. The first part describes the two approaches taken in the Opinion Summarization Pilot task and is structured as follows: in section 2, we present an overview of the system components, the processing steps and tools that were used in the two approximations we propose. The first approach suggested is explained in detail in Section 3, the tools used for this approximation in Section 3.1, and Section 3.2 contains the steps performed with the combination of these tools. In Section 4, the second approach for the Opinion Summarization Pilot task is explained, following the same structure as for the first approach. The evaluation and the experiments performed are reported in Section 5. Eventually, we present the main conclusions drawn from the experiments performed and the intended lines of future work. The second part of this paper deals with our participation in the Recognizing Textual Entailment track. In this part, section 2 contains the description of the system's components; in section 3, a study on possible constraints that could reduce the size of the processed corpus and the system's processing time are studied. Section 4 describes the experiments performed within the RTE track at TAC 2008 and the results obtained in the competition. Eventually, we present the conclusions drawn from the experiments.

PART I: Opinion Summarization Pilot

1. Introduction

The Summarization track was proposed with the aim of producing short coherent summaries of text. This track followed the Document Understanding Document[1] (DUC) conference's effort to have a common framework where summarization systems can be evaluated, compared and contrasted under the same conditions. The Opinion Summarization Pilot task consisted in generating summaries from blogs, according to specific opinion questions provided by the TAC organizers. Given a set of blogs from the *Blog06* collection and a list of questions from the Question Answering track, participating systems had to produce a summary that answered these questions. The questions generally required determining opinion expressed on 25 targets, each of which dealt with a single topic. Additionally, a set of text snippets were also provided, which contained the answers to the questions. These snippets were provided by real Question Answering systems, and opinion summarization systems could either use them or choose to perform themselves the retrieval of the answers to the questions in the corresponding blogs.

2. System components. Processing steps and tools.

In order to tackle the Opinion Summarization Pilot task, we considered the use of two different methods for opinion mining and summarization. The two approaches suggested differ mainly in the use of the optional text snippets provided by the TAC organization. Our first approach (the Snippet-driven Approach) used these snippets, whereas the second one (Blog-driven Approach) found the answers directly in the corresponding blogs.

The components, methods and tools involved in the system are summarized in Figure 1 and Figure 2.

The first phase, as shown in Figure 1, contains the question processing part. In order to extract the topic and determine the question polarity, we define question patterns. These patterns take into consideration the interrogation formula and extract the opinion words (nouns, verbs, adverbs, adjectives and their determiners). The opinion words are then classified in order to determine the polarity of the question, using the WordNet Affect (Strapparava and Valitutti, 2004) emotion lists, the emotion triggers resource, a list of four attitudes that we built, containing the verbs, nouns, adjectives and adverbs for the categories of criticism, support, admiration and rejection and two categories of value words (good and bad) taken from the opinion mining system in (Balahur and Montoyo, 2008 [2]).

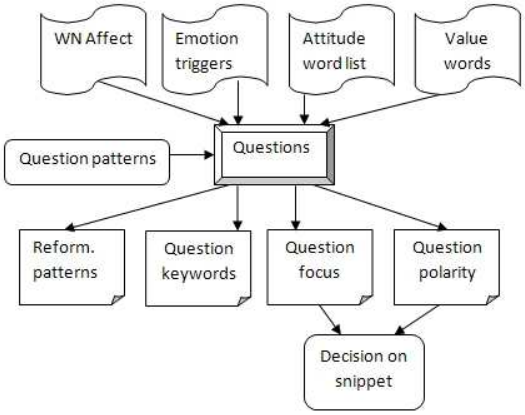

Figure 1. *The question processing stage*

[1] http://duc.nist.gov/

Examples of rules for the interrogation formula *"What reasons"* are:

1. *What reason(s) (.*?) for (not) (affect_verb + ing) (.*?)?*
2. *What reason(s) (.*?) for (lack of) (affect_noun) (.*?)?*
3. *What reason(s) (.*?) for (affect_adjective|positive|negative) opinions (.*?)?*

From this simple example, it can be seen how, by using patterns, we extracted the nouns, verbs, adjectives etc. that gave an indication of the question polarity. Further on, these indicators were classified according to the affect lists mentioned above.

The keywords of the question are determined by eliminating the stopwords. At the end of the question processing stage, we obtain, on the one hand, the reformulation patterns (that are eventually used to link and give coherence to the final summaries) and, on the other hand, the question focus, keywords and the question polarity. Depending on the focus/topic and polarity identified for each question, a decision on the further processing of the snippet was made, using the following rules:

1. If there is only one question made on the topic, determining its polarity is sufficient for making the correspondence between the question and the snippets retrieved; the retrieved snippet must simply obey the criteria that it has the same polarity as the question.
2. If there are two questions made on the topic and each of the questions has a different polarity, the correspondence between the question and the answer snippets can simply be done by classifying the snippets retrieved according to their polarity.
3. If there are two questions that have different focus but different polarities, the correspondence between the questions and the answer snippets is done using the classification of the answer snippets according to focus and polarity.
4. If there are two questions that have the same focus and the same polarity, the correspondence between the questions and the answer snippets is done using the order of appearance of the entities in focus, both in the question and in the possible answer snippet retrieved, simultaneously with the verification that the intended polarity of the answer snippet is the same as that of the question.

The categorization of questions into these four classes is decisive at the time of making the question – answer snippet correspondence, in the snippet/blog phrase processing stage. Details on these issues are given in what follows.

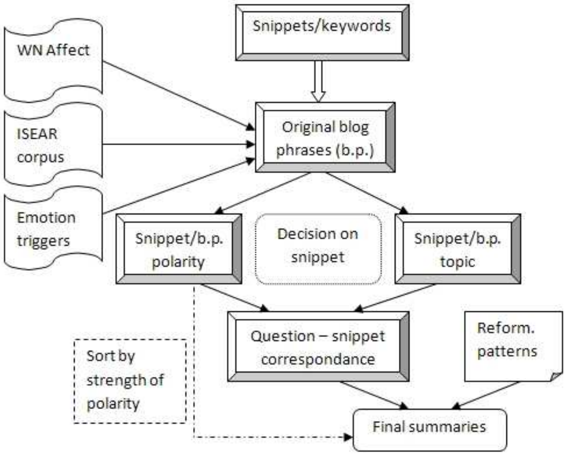

Figure 2. *The snippet/ blog phrase processing stage*

The second phase, as shown in Figure 2, contains the snippet (for the first approximation) and the blog phrases (for the second approximation) processing part. In the first approximation, the given answer-snippets constitute the basis for looking up the phrases these snippets were extracted from in the blogs collection. In the second approximation, we use the question keywords to determine the phrases from the blogs that could constitute answers to the questions. Further on, the blog original phrases or the blog retrieved phrases, respectively, are classified according to polarity, using the vector similarity with the set of vectors consisting of three distinct subsets. The first subset of vectors is built from the phrases in the ISEAR corpus (without stop words), one vector per statement and having the phrases classified according to the emotion it described. The second subset of vectors is built according to the WN Affect list of words in the joy, anger, sadness and fear. The third subset consists of the vectors of emotions from the emotion triggers resource, on each of the 4 categories that were considered also for building the vectors from WordNet Affect. For each of the blog phrase, we compute the similarity with all the vectors. Further on, each of the emotions is assigned a polarity – emotions from the FEAR, ANGER, SADNESS, DISGUST, SHAME, GUILT categories are assigned a negative polarity – and emotions from the JOY and SURPRISE categories we considered as positive. The final polarity score was computed as sum of the scores obtained in each of the vector similarity computations. The higher of the two scores – positive or negative - is considered as being the snippet polarity. In the second approximation, we also perform a sorting of the phrases retrieved, in descending order, according to their polarity scores. This is helpful at the time of building the final summary, whose length must not surpass a given limit. In the final phrases used in creating the summary we added, for coherence reasons, the reformulation patterns deduced using the question structure. Taken into consideration the number of characters limitation, we only included in the summary the phrases with high positive scores and those with high negative scores, completed with the reformulation patterns, until reaching the imposed character limit. Further details on each of the approaches and the tools used are given in section 3 and 4.

3. The Snippet-driven Approach

The basic idea behind this approach was to determine the focus and polarity in each of the given questions for a topic, determine the associated given answer snippet (by computing the snippet's polarity and focus), locate the whole sentence's snippet within the corresponding blog, and finally use patterns of reformulation from the questions' structure to bind together the snippets for the same polarity and focus to produce the final summary.

3.1. Tools used for the Snippet-driven Approach

Table 1 shows a brief description of the main resources which were used for the first approach. In the next Section, each of these resources will be explained in detail within the context in which it has been used for.

RESOURCE	TOOL	PURPOSE
Stemmer	Porter Stemmer	Locating sentences in orginal blogs.
Name Entity Recognizer	Freeling	Determining question's focus.
Text Similarity	Pedersen's Text Similarity package	Determining snippet's polarity and sentences in blogs.
Parser	Minipar	Filtering incomplete sentences out.
POS Tagger	TreeTagger	Changing verbs to impersonal speech style.
Emotion word lists	WordNet Affects	Determining question's and snippet's polarity.
Emotion word lists	ISEAR corpus	Determining snippet's polarity.
Emotion word lists	Emotions lists of attitudes	Determining question's polarity.
Emotion word lists	Emotion triggers	Determining question's and snippet's polarity.

Table 1. *Tools used for the snippet--driven and blog--driven approaches*

3.2. Methods used for the Snippet-driven Approach

The approach presented as our first approach for the Opinion Summarization Pilot task used the provided snippets to determine the original text fragments which answer the given questions. The first step was to determine the polarity of each question, extract the keywords from each of them and finally, build some patterns of reformulation, in order to provide the final summary a nature of an abstract rather than extracting the relevant sentences and putting all of them together. The polarity of the question was determined using a set of created patterns, whose goal was to extract for further classification the nouns, verbs, adverbs or adjectives indicating some kind of polarity (positive or negative). These extracted words, together with their determiners (in order to spot negations or polarity shiftings), were classified using the emotions lists in WordNet Affect (Strapparava and Valitutti, 2005), jointly with the emotions lists of attitudes, triggers resource (Balahur and Montoyo, 2008 [1]), four created lists of attitudes, expressing criticism, support, admiration and rejection and two categories for value (good and bad), taking for the opinion mining systems in (Balahur and Montoyo, 2008 [2]). Moreover, the focus of each question was automatically extracted by means of the name's entity recognizer module of Freeling[2] in order to know whether or not all the questions within the same topic had the same focus, as well as, be able to decide on later which text snippet belongs to each question.

Regarding the given text snippets, we also computed their polarity and their focus. The polarity was calculated as a vector similarity between the snippets and vectors constructed from the list of sentences contained in the ISEAR corpus (Scherer and Wallbot, 1997), WordNet Affect emotion lists of anger, sadness, disgust and joy and the emotion triggers resource, using Pedersen's Text Similarity Package.[3]

Concerning the blogs, we converted the HTML blogs into plain text, removing all unnecessary tags which contained any information at all and splitting the blog into individual sentences. A matching between blogs' sentences and text snippets was performed so that a preliminary set of potential meaningful sentences was recorded to further processing. To achieve this, snippets not literally contained in the blogs were tokenized and stemmed using Porter's Stemmer,[4] and stop words were removed in order to find the most similar possible sentence associated with it. Afterwards, by means of the same Pedersen Text Similarity Package as for computing the snippets' polarity, we computed the similarity between the given snippets and this created set of potential sentences, and we extracted the complete sentences of the blogs to which each snippet was related, extracting the focus for each blog phrase sentence as well. Due to the fact that information might be repeated across different snippets, we filter this using a naïve attempt to remove redundant sentences. Once we obtained the possible answers, we used Minipar[5] to filter incomplete sentences out. The reason for performing this opperation was to avoid sentence chunks without meaning from becoming part of the final summary. An example of a complete sentence would be *"that just might begin to reduce our oil dependence"*, and two examples of incomplete sentences analysed by means of Minipar can be seen in Figure 3.

```
PUBLIC INTEREST.CO.UK
> (
1     (PUBLIC     ~ A   2     mod   (gov INTEREST.CO.UK))
2     (INTEREST.CO.UK   ~ N   *     )
)
>
posted by Jeff McIntire-Strasburg @ 8:18 PM Comments Trackback
> (
1     (posted     ~ U   *     punc)
```

[2] http://garraf.epsevg.upc.es/freeling/

[3] http://www.d.umn.edu/~tpederse/text-similarity.html

[4] http://tartarus.org/~martin/PorterStemmer/

[5] http://www.cs.ualberta.ca/~lindek/minipar.htm

```
E1    (()    U    *     )
2     (by   ~ Prep      E1    p)
3     (Jeff ~ U   6     lex-mod     (gov Jeff McIntire-Strasburg))
4     (McIntire   ~ U   6     lex-mod     (gov Jeff McIntire-Strasburg))
5     (-    ~ U   6     lex-mod     (gov Jeff McIntire-Strasburg))
6     (Strasburg  Jeff McIntire-Strasburg N     2     pcomp-n     (gov
by))
7     (@    ~ U   E1    punc)
8     (8:18 ~ U   E1    punc)
9     (PM   ~ U   10    lex-mod     (gov PM Comments))
10    (Comments   PM Comments N     11    nn    (gov Trackback))
11    (Trackback  ~ N   E1    )
)
>
```

Figure 3. *Minipar analysis for two incomplete sentence.*

Having computed the polarity for the questions and text snippets, and having set out the final set of sentences to produce the summary with their focus, we bound each sentence to its corresponding question, and we grouped all sentences which were related to the same question together, so that we could generate the language for this group, according to the patterns of reformulation that were created for each question. Finally, the speech style was changed to an impersonal one, in order to avoid the presence of directly expressed opinion sentences. A POS-tagger tool (TreeTagger[6]) was used to identify third person verbs and change them to a neutral style. A set of rules to identify pronouns was created, and they were also changed to the more general pronoun "they" and its corresponding forms, to avoid personal opinions. Figures 4 shows an example of an original blog sentence whose verbs and pronouns have to be changed (sentence at the top), and the sentence once changed (sentence at the bottom).

```
I am sure YouTube has already found itself on the blacklist of most
school Internet filters.

they are sure YouTube have already found itself on the blacklist of
most school Internet filters.
```

Figure 4. *Pronoun and verb change.*

4. The Blog-driven Approach

4.1. Tools used for the Blog-driven Approach

Since the tools used in the second approach are the same as those used in the first approximation, the main resources employed for the second approach can be found in Table 1.

4.2. Methods used for the Blog-driven Approach

The second approach had as starting point determining the focus, keywords, topic and polarity in each of the given questions. The processing of the question is similar to the one performed for the first approximation. Starting from the focus, keywords and topic of the question, we sought sentences in the blog collection (previously processed as described in the first approximation) that could constitute possible answers to the questions, according to their similarity to the latter. The similarity score was computed with Pedersen's Text Similarity Package.

[6] http://www.ims.uni-tuttgart.de/projekte/corplex/TreeTagger/

The snippets thus determined underwent dependency parsing with Minipar and only the sentences which contained subject and predicate were kept, thus ensuring the elimination of some of the present “noise” (such as section titles, dates, times etc.). The remaining snippets were classified according to their polarity, using the similarity score with respect to the described emotion vectors. The direct language style was changed to indirect speech style. In that manner, all direct phrases such as *“I liked the show”* changed to *“They liked the show”*, and *“I am surprised by their coffee”* to *“They are surprised by their coffee”*. The reformulation patterns that were deduced using the questions’ structure were added to bind together the snippets and produce the final summary, concatenating the snippets with the added reformulations. Since the final length of the summary could easily overpass the imposed limit, we sorted the snippets using their polarity strength (the higher the polarity score – be it positive or negative- the higher the rank of the snippet), and included the reformulated snippets in descending order until the final limit was reached.

5. Evaluation

5.1. Evaluation results

45 runs were submitted by 19 teams for evaluation in the TAC 2008 Opinion Pilot task. Each team was allowed to submit up to three runs, but finally, due to the difficulty involved in the evaluation of such a task, only the first two runs of each team was evaluated, leading to 36 runs being evaluated.

Table 2 shows the final results obtained by the first two runs we submitted for evaluation in the TAC 2008 Opinion Pilot. The column numbers stand for the following information:

1. summarizerID (our Run 1 had summarizerID 8 and Run 2 had summarizerID 34)
2. Run type: "manual" or "automatic"
3. Did the run use the answer snippets provided by NIST: "Yes" or "No"
4. Average pyramid F-score (Beta=1), averaged over 22 summaries
5. Average score for Grammaticality
6. Average score for Non-redundancy
7. Average score for Structure/Coherence (including focus and referential clarity)
8. Average score for Overall fluency/readability
9. Average score for Overall responsiveness

1	2	3	4	5	6	7	8	9
8	automatic	Yes	0.357	4.727	5.364	3.409	3.636	5.045
34	automatic	No	0.155	3.545	4.364	3.091	2.636	2.227

Table 2. *Evaluation results.*

Further on, we will present the system performances with respect to all other teams, first as an overall classification (Table 3) and secondly, taking into consideration whether or not the run used the optional answer snippets provided by NIST (Table 4). In Table 3, the numbers in columns 4, 5, 6, 7, 8 and 9 correspond to the position within the 36 evaluated submissions. In Table 4, the numbers in columns 4, 5, 6, 7, 8 and 9 correspond to the position within the 17 submissions that used the given optional answer snippets (in case of Run 1) and the position within the 19 submissions evaluated that did not use the provided answer snippets.

1	2	3	4	5	6	7	8	9
8	automatic	Yes	7	8	28	4	16	5
34	automatic	No	23	36	36	13	36	28

Table 3. *Classification results (overall comparison).*

1	2	3	4	5	6	7	8	9
8	automatic	Yes	7	15	14	2	11	5
34	automatic	No	9	19	19	6	19	14

Table 4. *Classification results (comparison with systems using/not using answer snippets).*

5.2. Analysis of results.

As it can be noticed from the results table, our system performed well regarding Precision and Recall, the first run begin classified 7th among the 36 evaluated runs as far as F-measure. As far as the structure and coherence are concerned, the results were also good, placing Run 1 in fourth position among the 36 evaluated runs. Also worth mentioning is the good performance obtained as far as the overall responsiveness is concerned, where Run 1 ranked 5th among the 36.

When comparing our approaches separately, in both cases, they did not perform very well with respect of the non-redundancy criterion, nor the grammaticality one. We thought of the idea of using a Textual Entailment engine to identify and remove redundant sentences, but due to time constraints we finally could not use it, and we opted for removing only those sentences that were exactly the same. Regarding the grammaticality, the results show that we should improve the methods for filtering non-relevant information out, avoiding that the "noise" information appears in the final summary. However, an interesting thing that is worth mentioning as far as the results obtained are concerned, is that the use of reformulation patterns, in order to generated sentences for completing the summaries, has been appropriate, leading to very good positions according to the structure/coherence criterion. We are also quite satisfied with the responsiveness and the F-score of the summaries, because even though we did not use the snippets provided by NIST for the second approach, we were able to locate the most relevant information within blogs, only by using the similarity between the sentences and the keywords extracted from the questions, and using polarity classification as relevance criteria for the inclusion of the retrieved text in the final summary. For the first approach we suggested, results showed well-balanced among all the criteria evaluated, except for non redundancy and grammaticality. For the second approach, aside from the previously mentioned criteria, we have to take into consideration the fluency of the summary as well, for future research. An interesting and important test that we must perform is on the influence of the polarity classification and computation of the polarity strength of the snippets, which was the selection criterion for including or not the snippet in the final summary.

6. Conclusions

With the participation in the Opinion Pilot Task we could, on the one hand, test a general opinion mining system within a multi-perspective question answering framework and, on the other hand, test the importance of polarity strength as to what answer relevance to such questions is concerned within a summary. We could also study the influence that different Natural Language Processing resources, such as parsers, similarity detection tools or name entity recognizers have on the summarization task, and measure to what extent they are useful to detect non-relevant information or to select important content to belong to the final summary. Although our approaches were not very complex – based on question patterns for question focus and polarity detection and language generation, similarity for retrieval and polarity classification, in performing the given task we were confronted with different problems to which the solutions found could solve the majority of issues: using dependency parsing to eliminate some of the present noise, such as titles, dates, etc. or performing POS-tagging to change the 3rd person formulations. As the results are encouraging, the participation in the competition is a good starting point for the building of a system that is capable of interpreting and opinion question, retrieve possible answers to it, filtering them, and eventually presenting a correct and concise answer.

PART II: The Recognizing Textual Entailment Track

1. Introduction

The goal of the RTE track is to develop systems that recognize when one piece of text entails another. The RTE track at TAC continued the efforts of the PASCAL RTE Challenges (Giampiccolo et al., 2007). With our participation in RTE, we aim to test and evaluate the developments and improvements of our previous RTE system (Ferrández et al., 2007). Within the RTE track, we participated in the 2-way evaluation. The reason motivating our decision to participate only in the 2-way evaluation was given, on the one hand, by the fact that our RTE system does not deal with logic representation On the other hand, finding an uncertainty range of similarity values that show when the system is not able to determine true or false entailment is a complex task due to the scarcity of this kind of training examples.

2. System components

Figure 4 depicts an overview of our RTE-system's work flow. The figure shows the system at a glance, drawing the resources that are consumed by each component.

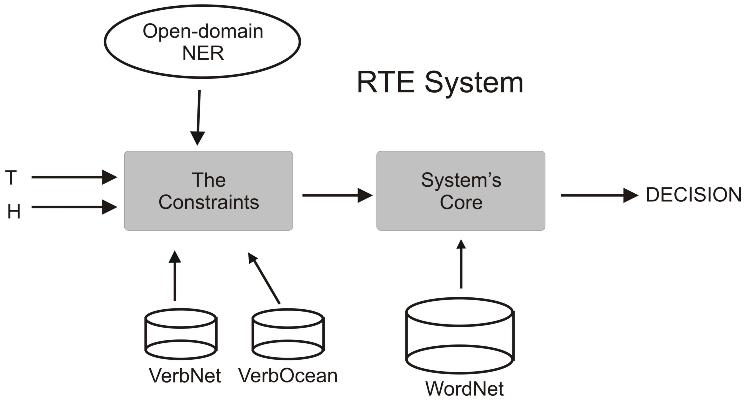

Figure 4. *RTE System's Core.*

The main objective of the research presented herein was to build a Textual Entailment (TE) system that would serve as the base system performing the remaining inferences considered in the task of textual entailment recognition (i.e the system's core). The most reasonable way to create it is by means of lexical-standard measures capable of detecting TE relations regardless of the language or domain of the texts. These measures have already been used for many researchers obtaining very promising results (see the (Giampiccolo et al., 2007) report on the last RTE Challenge). However, the lack of semantic knowledge raises doubts about the robustness of systems using only these kinds of measures.

A wide variety of lexical-standard measures[7] were considered. We performed a study on what the most significant for the RTE task are according to the information gain that they provide to a machine learning classifier and over the RTE training corpora available. The selected measures were processed by the SVM algorithm developed in Weka.[8] Some considered measures are the Levenshtein distance, the Smith-Waterman

[7] For some measures we use their implementation provided by the SimMetrics library (http://www.dcs.shef.ac.uk/~sam/simmetrics.html)

[8] http://www.cs.waikato.ac.nz/ml/weka/

algorithm and the Cosine similarity, to name but a few. Due to space constraints, these measures have not been explained in detail, but we kindly redirect the reader to (Ferrández et al., 2008).

Apart from the aforementioned lexical-standard measures, we would like to point out two similarity measures that are also integrated into the system's core. We decided to add these measures to the system's core due to their being easily adaptable to other languages and their being widely used in similarity tasks.

- *IDF specificity*: we determine the specificity of a word using the well-known inverse document frequency (IDF). We derive the documents frequencies from the collections used within the Cross-Language Evaluation Forum[9] (CLEF), in particular the LA Times 94 and Glasgow Herald 95 collections (169,477 documents). This metric was considered as an individual feature to decide the entailment as follows:

- *JWSL*: in order to discover word meaning relations that are not able to be detected directly from orthographic derivations, we use WordNet. Relations such as synonymy, hyperonymy, and semantic paths that connect two concepts are exploited, obtaining similarity and relatedness measures between two words. To achieve this, we have used the Java WordNet Similarity Library[10] (JWSL), which implements some of the most commons semantic similarity measures. We created a procedure that derives a score (the maximum score obtained from all similarity measures implemented in JWSL) from the best relations between the words of the hypothesis and the text. This score is also considered as a system feature.

3. RTE System's Constraints

Two constraints were added to the system’s core. These constraints show the system’s behaviour when some semantic knowledge is taken into account. Moreover, we evaluate whether the constraints reduce the size of the corpus processed and consequently the system's processing time. These constraints are processed prior to the computation of the inferences belonging to the system's core; thus, if the T-H pair successfully passes them it will be considered as a possible true entailment pair.

The Importance of being Named Entity. It is based on the detection, presence and absence of Named Entities (NEs), measuring the importance of the presence or absence of an entity (e.g. when there is an entity in the hypothesis but the same entity is not present in the text). This idea comes from the work presented in (Rodrigo et al., 2008), where the authors successfully build their system only using the knowledge supplied by the recognition of NEs. In our case, we set a constraint previous to the system's core inferences that only considers as candidate entailment pairs those in which the entities in H also appear in T. In our experiments, we use the NERUA system (Kozareva et al., 2007), an open domain NE recognizer. A partial entity matching was considered (i.e. ”George Bush”, ”George Walker Bush”, ”G. Bush” and ”Bush” are considered as the same entity). Unfortunately, reasoning about acronyms, date expansion, metonymy and location/demonymy was not developed at the current state of the system. Subsequent work on this area will be characterized by the addition of this sort of reasoning.

The importance of being Verb. Verbs are very important particles to the sentence meaning. Therefore, with this constraint we attempt to measure the relatedness between the H's verbs and the T's verbs. To do this, we created two wrappers in Java for the VerbNet[11] and VerbOcean[12] resources in such a way that if every verb in the hypothesis (auxiliar verbs are not considered) can be related to one or more verbs in the text, the pair will successfully pass this constraint. Two verbs are related whether: (i) they have the same lemma or are

[9] http://www.clef-campaign.org/

[10] http://grid.deis.unical.it/similarity/

[11] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html

[12] http://demo.patrickpantel.com/Content/verbocean/

synonyms considering WordNet, (ii) they belong to the same VerbNet class or a subclass of their classes, or (iii) there is a relation in VerbOcean[13] that connects them.

We also studied to integrate these constraints into the system as new features for the SVM algorithm. However, it did not report any improvement in the final results. Furthermore, we decided to consider these inferences as previous constraints, since (although the results decrease slightly) the corpus as well as the processing time are strongly reduced.

4. RTE experiments and results

Table 5 shows the results obtained for both the development and test corpus, and for every experiment carried out. The test corpus was provided by the RTE-TAC organizers; however, no development corpora were supplied. Thus, we have used the development and test corpora of the last editions of the RTE Challenges in order to train our system. Several options were considered, and although training the system with the corpora belonging to the last RTE-3 challenge obtained a slight increase in accuracy, we decided to use the RTE-2 and RTE-3 development & test corpora, due to the fact that these corpora provided a large variety of examples, which is desirable for new incoming data.

	Run	**Acc.**		**Run**	**Acc.**
Development	System's Core (SC)	0.649	Test	System's Core (SC)	0.608
	SC+ENT	0.634		SC+ENT	0.599
	SC+ENT+VERB	0.624		SC+ENT+VERB	0.594

Table 5. *DLSIUAES RTE-system results within the 2-way entailment evaluation.*

We carried out three experiments: (1) *System's Core (SC)*, processing the inferences shown in part II section 2; (2) *SC+ENT*, adding the first constraint regarding the presence or absence of NEs; and (3) *SC+ENT+VERB*, also considering the constraint about H-to-T verbs relations.

As anticipated, the addition of both constraints causes a slight decrease in accuracy. However, at this point, we have to assess the benefits of these constraints. Figure 5 draws the ratio of the development and test corpus that the system did not have to process because of the entailment pair did not pass one of the constraints.

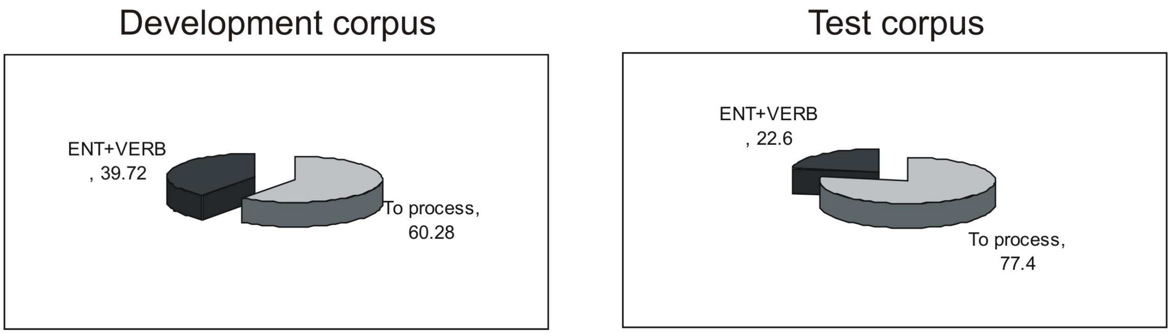

Figure 5. *Percentage of corpus not processed due to the constraints.*

More importantly, the pairs that did not pass the constraints were annotated as false entailment. These pairs obtained an accuracy rate of 62% and 68.14% in correct no entailment detection for the development and test corpora respectively.

Future work proposals for our RTE system would be those related to the addition of deeper semantic analysis. For instance, a role labelling module based on FrameNet[14] and Shalmaneser[15] in order to detect frame and frame elements relations between the entailment pairs, is our priority subsequent work.

[13] The VerbOcean's relations considered are: similarity, strength and happens-before.

5. Conclusions

The RTE system presented in this paper tackles the entailment phenomenon from two different points of view. First, we build the system's core by means of several lexical measures and further on, we add some semantic constraints that we think are appropriated for the entailment recognition. The reason for creating this core was given by (i) the fact that the integration of more complex semantic knowledge is a delicate task and it would be easier if we had a solid base system; and (ii) although the proposed core needs some language dependent tools (e.g. lemmatizer, stemmer), it could be easily ported to other languages. Results point out that promising accuracy is reached by the system's core. Regarding the semantic constraints, although they do not obtain better results they dramatically reduce the data processed by the system and consequently its total processing time.

Acknowledgements

This research has been supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01) and partially subsidized by the QALL-ME consortium, 6th Framework Research Programme of the European Union (EU), contract number: FP6-IST-033860.

References

1. Balahur, A. and Montoyo, A. [1]. An Incremental Multilingual Approach to Forming a Culture Dependent Emotion Triggers Database. In Proceedings of the 8th International Conference on Terminology and Knowledge Engineering (TKE 2008), Copenhagen.
2. Balahur, A. and Montoyo, A. [2]. Multilingual Feature--driven Opinion Mining and Summarization from Customer Reviews. In Lecture Notes in Computer Science 5039, pg. 345-346.
3. Ferrández, O. , Micol, D., Muñoz, R. and Palomar, M. "A Perspective-Based Approach for Solving Textual Entailment Recognition". In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June, 2007, pp. 66-71.
4. Ferrández, O., Muñoz, R. and Palomar, M. "A lexical-semantic approach to AVE". In Working Notes of the CLEF 2008 Workshop, September, 2008.
5. Giampiccolo, D., Magnini, B., Dagan, I. and Dolan, B. "The Third PASCAL Recognizing Textual Entailment Challenge".In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June, 2007, pp. 1-9.
6. Kozareva, Z. , Ferrández, O., Montoyo, A. Muñoz, R."Combining data--driven systems for improving Named Entity Recognition". In Data and Knowledge Engineering, 2007, vol. 61, No 3, pp. 449-466.
7. Rodrigo, Á., Peñas, A. and Verdejo, F. "UNED at Answer Validation Exercise 2007". In Advances in Multilingual and Multimodal Information Retrieval, 8th Workshop of the Cross-Language Evaluation Forum, 2008, pp. 404-409.
8. Scherer, K. and Wallbott, H.G. The ISEAR Questionnaire and Codebook, 1997.
9. Strapparava, C. and Valitutti, A. "WordNet-Affect: an affective extension of WordNet". In Proceedings ofthe 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, May 2004, pp. 1083-1086.

[14] http://framenet.icsi.berkeley.edu/

[15] http://www.coli.uni-saarland.de/projects/salsa/shal/