

# Uncovering discourse relations to insert connectives between the sentences of an automatic summary

Sara Botelho Silveira and António Branco

University of Lisbon, Portugal,  
{sara.silveira,antonio.branco}@di.fc.ul.pt,  
WWW home page: <http://nlx.di.fc.ul.pt/>

**Abstract.** This paper presents a machine learning approach to find and classify discourse relations between two unseen sentences. It describes the process of training a classifier that aims to determine (i) if there is any discourse relation among two sentences, and, if a relation is found, (ii) which is that relation. The final goal of this task is to insert discourse connectives between sentences seeking to enhance text cohesion of a summary produced by an extractive summarization system for the Portuguese language.

**Keywords:** discourse relations, discourse connectives, summarization

## 1 Motivation

An important research issue in which there remains much room for improvement in automatic text summarization is text cohesion. Text cohesion is very hard to ensure specially when creating summaries from multiple sources, as their content can be retrieved from many different documents, increasing the need for some organization procedure.

The approach presented in this paper aims to insert discourse connectives between sentences seeking to enhance the cohesion of a summary produced by an extractive summarization system for the Portuguese language [17]. Connectives are textual devices that ensure text cohesion, as they support the text sequence by signaling different types of connections or discourse relations among sentences. It is possible to understand a text that does not contain any connective, but the occurrence of such elements reduces the cost of processing the information for human readers, as they explicitly mark the discourse relation holding between the sentences, thus acting like guides in the interpretation of the text. The assumption in this work is that relating sentences that are retrieved from different source texts can produce a more interconnected text, and thus a more easier to read summary.

Marcu et al. (2002) noted that “discourse relation classifiers trained on examples that are automatically extracted from massive amounts of text can be used to distinguish between [discourse] relations with accuracies as high as 93%,”

even when the relations are not explicitly marked by cue phrases” [9]. Following the same research line, this paper presents a machine learning approach relying on classifiers that predict the relation shared by two sentences. Considering two adjacent sentences, the final goal is to insert, between those sentences, a discourse connective that stands for the discourse relation found between them – including possibly the phonetically null one.

The procedure is composed by two phases. The first phase – *Null vs. Relations* – determines if two adjacent sentences share a discourse relation or not. If a relation has been found, the second phase – *Relations vs. Relations* – is applied, aiming to distinguish which is the discourse relation both sentences share. Based on this relation a discourse connective is retrieved from a previously built list to be inserted between those sentences. Consider for example the following sentences.

- S<sub>1</sub> *O custo de vida no Funchal é superior ao de Lisboa.*  
 The cost of living in Funchal is higher than in Lisbon.  
 S<sub>2</sub> **No entanto**, *o Governo Regional nega essa conclusão.*  
 However, the Regional Government denies this conclusion.

These two sentences are related by the discourse connective *no entanto* (“however”), which expresses that the two sentences convey some adversative information. Hence, it is possible to say that these sentences entertain a relation of COMPARISON-CONTRAST-OPPOSITION, based on the discourse connective that relates them. The following example runs through the complete procedure.

1. Retrieve two adjacent sentences.  
*O custo de vida no Funchal é superior ao de Lisboa*  
 The cost of living in Funchal is higher than in Lisbon.  
*O Governo Regional nega essa conclusão.*  
 The Regional Government denies this conclusion.
2. Find the discourse relation.  
 Apply model *Null vs. Relations* → “Yes” = both sentences share indeed a discourse relation.  
 Apply model *Relations vs. Relations* → relation class = COMPARISON-CONTRAST-OPPOSITION
3. Look for the connective to insert.  
 A random connective is obtained in the list for the class COMPARISON-CONTRAST-OPPOSITION → retrieved: *no entanto*.
4. Insert the discourse connective between the two sentences.  
*O custo de vida no Funchal é superior ao de Lisboa*  
 The cost of living in Funchal is higher than in Lisbon.  
*No entanto*, *o Governo Regional nega essa conclusão.*  
 However, the Regional Government denies this conclusion.

The remainder of this paper is structured as follows. Section 2 overviews previous works on finding discourse relations in text and details the approach

pursued in this work; and Section 3 points out some future work directions, based on the conclusions drawn.

## 2 Uncovering discourse relations

The intent of the majority of the studies that address discourse relations is to recognize ([9], [5], [2], [7], [10], [19]) and classify discourse relations in unseen data ([20], [12], [11]).

Other works ([8], [1], [3]) approach this problem with different goals. Louis et al. aim to enhance content selection in single-document summarization [8]. Biran and Rambow are focused in detecting justifications of claims made in written dialog [1]. Feng et al. seek to improve the performance of a discourse parser [3].

Despite of their different goals, these studies follow a common approach to find and classify discourse relations in text, that is a machine learning techniques over annotated data. The task is to learn how and which discourse relations are explicitly – by means of cue phrases – or implicitly expressed on human annotated data. In the approach presented in this paper, this task is reverted. The classification of the discourse relation will be used to determine a discourse connective to be inserted between a given pair of adjacent sentences.

In order to build a classifier that decides which discourse relation is holding between two sentences, there are several decisions at stake: the initial corpus, the features to be used, the training and testing datasets, and the classification algorithm. The remainder of this section discusses these decisions.

### 2.1 Discourse corpus

In order to feed the classifiers, a corpus that explicitly associates a discourse relation to a pair of sentences was created semi-automatically, relying on a corpus of raw texts and a list of discourse connectives.

The list of Portuguese discourse connectives was built by a human annotator who started by translating list provided by the English *Penn Discourse TreeBank (PDTB)* [14] [13]. After a first inspection to the raw corpus and taking into account the convenience of this task, some adjustments were made to this list, resulting in a final list that was used to create the discourse corpus. Table 1 shows an example of a connective for each class.

Prasad et al. (2008) state that discourse connectives have typically two arguments: ARG<sub>1</sub> and ARG<sub>2</sub>. Also, they concluded that the typical structure in which the three elements are combined is ARG<sub>1</sub> <CONNECTIVE> ARG<sub>2</sub>. The following example shows two sentences with this typical structure, where S<sub>1</sub> maps to ARG<sub>1</sub> and S<sub>2</sub> maps to ARG<sub>2</sub>, with the connective “but” being included in ARG<sub>2</sub>.

S<sub>1</sub> *Washington seguiu Saddam desde o início.*

Washington followed Saddam from the beginning.

S<sub>2</sub> ***Mas** a certa altura as comunicações com Clinton falharam.*

But at some point communications with Clinton failed.

**Table 1.** Examples of discourse connectives by class.

Class	Connective	Translation
COMPARISON-CONTRAST-OPPOSITION	mas	but
COMPARISON-CONCESSION-EXPECTATION	apesar de	although
COMPARISON-CONCESSION-CONTRA-EXPECTATION	como	as
CONTINGENCY-CAUSE-REASON	pois	because
CONTINGENCY-CAUSE-RESULT	então	hence
CONTINGENCY-CONDITION-HYPOTHETICAL	a menos que	unless
CONTINGENCY-CONDITION-FACTUAL	se	if
CONTINGENCY-CONDITION-CONTRA-FACTUAL	caso	if
TEMPORAL-ASYNCHRONOUS-PRECEDENCE	antes de	before
TEMPORAL-ASYNCHRONOUS-SUCCESSION	depois de	after
TEMPORAL-SYNCHRONOUS	enquanto	until
EXPANSION-RESTATEMENT-SPECIFICATION	de facto	in fact
EXPANSION-RESTATEMENT-GENERALIZATION	em conclusão	in conclusion
EXPANSION-ADDITION	adicionalmente	additionally
EXPANSION-INSTANTIATION	por exemplo	for instance
EXPANSION-ALTERNATIVE-DISJUNCTIVE	ou	or
EXPANSION-ALTERNATIVE-CHOSEN ALTERNATIVE	em alternativa	instead
EXPANSION-EXCEPTION	caso contrário	otherwise

*CETEMPública* [16] is a corpus built from excerpts of news from *Público*, a Portuguese daily newspaper. This corpus was analyzed to find pairs of sentences complying with this structure. The composition of the discourse corpus is defined by triples as such ( $ARG_1$ ,  $ARG_2$ , *DiscourseRelation*). So, after gathering the sentence pairs, a classification is required for the discourse relation holding between each pair of sentences. [13] argue that this typical structure is the minimal amount of information needed to interpret a discourse relation. Then, each pair was classified with the class of the discourse connective that links its sentences together. Also, the connective is removed from the sentence defined as  $ARG_2$ .

Finally, taking into account the goal of the task presented in this paper, when considering two adjacent sentences, those can share a discourse relation or not. Thus, pairs of adjacent sentences that do not have any discourse relation, that is that are not linked by any of the connectives considered, have also been retrieved. All the pairs that do not contain any connective linking them were classified with the NULL class, stating that there is no relation between the sentences.

This way a discourse annotated corpus has been built relating a pair of sentences and their respective discourse relation. This corpus was then used to create the datasets used to train and test the classifiers.

## 2.2 Experimental settings

Experimental settings comprise the features, the datasets and the classification algorithms that were used to train the classifiers.

*Features.* Considering the task at hand, the features are expected to reflect the properties that could express the discourse relation holding between the two arguments in the relation ( $\text{ARG}_1$  and  $\text{ARG}_2$ ).

In order to find the best configuration, for the experiments, several features were tested. Considering the structure of the discourse corpus, the most straightforward approach would be to use both sentences ( $\text{ARG}_1$  and  $\text{ARG}_2$ ) to train the classifier.

Previous works ([9], [6], [20], [7], [8]) essayed different types of features to classify discourse relations, including contextual features, constituency features, dependency features, semantic features and lexical features. The presented approach is inspired in the one of Wellner et al. that reported high accuracy when using a combination of several lexical features [20].

In a sentence, the verb expresses the event so it can constitute a relevant information in helping to distinguish between different relations. Considering a specific relation, different pairs of sentences sharing that relation might have different verbs, although they could have the same discourse connective. This discourse connective typically requires the same verb inflections, not necessarily the same instance of the verb. Thus, instead of using the verb in each sentence, the verb inflections of each sentence were used.

Another feature is related to the context in which the discourse connective appears. Thus, a context window surrounding the occurrence of the discourse connective will be used. A six-word context window surrounding the location where the discourse connective occurs in the discourse relation is considered, where three words are the last three words of  $\text{ARG}_1$  and the other three words are the first three words of  $\text{ARG}_2$ .

In addition, three more features were used to improve the identification of the tiny differences across discourse relations. These features include all the adverbs, conjunctions and prepositions found in each of the sentences. Conjunctions link words, phrases, and clauses together. Adverbs are modifiers of verbs, adjectives, other adverbs, phrases, or clauses. An adverb indicates manner, time, place, cause, or degree, so that it may help unveiling the grammatical relationships within a sentence or a clause. A non functional, semantically loaded preposition usually indicates the temporal, spatial or logical relationship of its object to the rest of the sentence. All these words can constitute clues to better identify the discourse relation between two unseen sentences, so they can help to enhance the accuracy of the classifier.

*Datasets.* The discourse corpus distribution indicates that it is highly uneven, containing some very big classes (e.g. NULL) and at the same time some very small ones (e.g. CONTINGENCY-CONDITION-FACTUAL). Taking this into account, all the experiments were based on even training datasets, that is the datasets always contain the same number of examples for each class.

Moreover, the training procedure was split in two phases. In the first training phase, the goal is to train a classifier that aims to identify whether the sentences share a discourse relation or not (named *Nulls vs. Relations*). Thus, the first

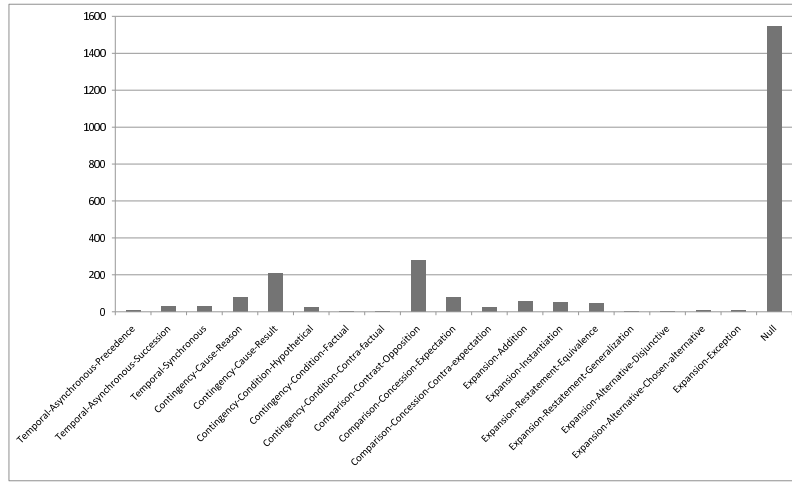
dataset includes pairs from all the discourse classes – assigned as RELATION – and pairs assigned with the NULL class – assigned as NULL.

After uncovering that two sentences share indeed a discourse relation, the second training phase (named *Relations vs. Relations*) seeks to find which is that discourse relation. The second dataset will only include the pairs assigned with a specific discourse class (the NULL pairs are not included).

In what concerns the testing dataset, it will remain imbalanced as to reflect the normal distribution of discourse relations in a corpus.

Figure 1 illustrates the distribution of the testing dataset in the *Null vs. Relations* training phase, while Figure 2 shows the classes distribution in the *Relations vs. Relations* phase.

**Fig. 1.** Distribution of the classes in the testing dataset for *Null vs. Relations*.



*Classification algorithms.* There are several algorithms that have been more frequently used in Natural Language Processing tasks.

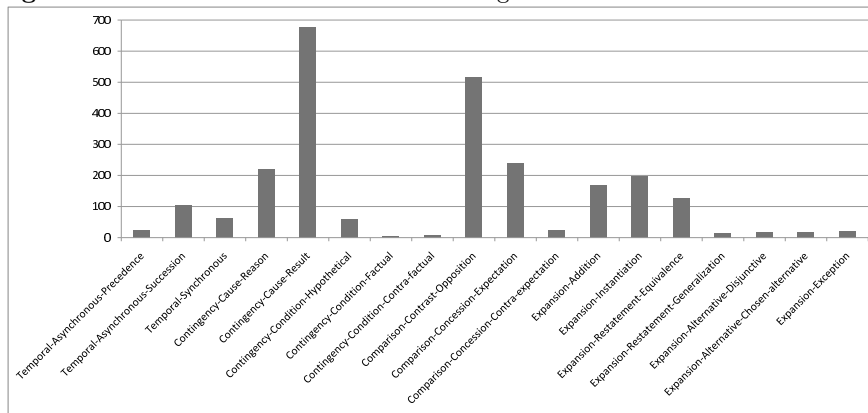
*Naïve Bayes* [4] is a probabilistic classifier, which algorithm assumes independence of features as suggested by Bayes' theorem. Despite its simplicity, it achieves similar results obtained with much more complex algorithms.

*C4.5* [15] is a decision tree algorithm. It splits the data into smaller subsets using the information gain in order to choose the attribute for splitting the data. In short, decision trees hierarchically decompose the data, based on the presence or absence of the features in the search space.

Finally, *Support Vector Machines (SVM)* [18] is an algorithm that analyzes data and recognizes patterns. The basic idea is to represent the examples as points in space, making sure that separate classes are clearly divided. *SVM* is a binary classifier, specially suitable for two-class in classification problems.

All these algorithms were used in the experiments reported in this paper.

**Fig. 2.** Distribution of the classes in the testing dataset for *Relations vs. Relations*.



## 2.3 Results

The classifiers trained aim to learn how to distinguish which is the discourse relation that two unseen sentences share, if any.

The first assumption of such a task would be to take into account all the classification classes at the same time. This means that, in this case, 19 classes would be taken into account, including all the 18 possible discourse classes plus the NULL class. However, as the corpus from which the datasets were built is highly imbalanced between all the 19 classes (as the distribution of the testing datasets suggest – cf. Figures 1 and 2), the classifier training procedure was divided in two phases, as already stated. *Null vs. Relations* determines if the sentences share a discourse relation or not. *Relations vs. Relations* discovers which is the discourse relation, already knowing that both sentences share one.

All the experiments reported were obtained using *Weka* workbench [21].

*Null vs. Relations.* The experiment procedure defines first the training and testing datasets used. The training dataset includes the same number of examples from all classes divided in a binary classification problem. The classifier will have then to decide if two sentences share a discourse relation (“yes”) or if they do not (“no”). It contains 5,000 pairs evenly divided in both these classes. Yet, the testing dataset contains 2,500 pairs reflecting the normal distribution of the discourse relations in a corpus by remaining imbalanced (cf. Figure 1).

This first experiment aims mainly to identify which features improve the classifier accuracy, so that several combinations of features were used.

The results were then obtained using the most common classification algorithms (previously discussed).

The configuration for the first experiment is the following:

- Testing dataset – 2,500 pairs from the testing dataset
- Training dataset – 5,000 pairs split in half for each class, NULL and RELATION

- Features essayed:
  1. Complete sentences: ARG<sub>1</sub> and ARG<sub>2</sub>
  2. Verb inflections
  3. Verb inflections and 6-word context window
  4. Verb inflections, 6-word context window, adverbs, prepositions, and conjunctions (*inf-6cw-adv-prep-cj*)
- Algorithms:
  - Naïve Bayes
  - C4.5 decision tree
  - Support Vector Machines (SVM)

In order to understand the results obtained when varying the features and the algorithms, a baseline for this task must be considered. The baseline assigns the most frequent class to all the instances. The most frequent class in the dataset is the NULL class. So, when assigning always the NULL class to the instances occurring in the test set, it is possible to achieve an accuracy of 61%, being this value the baseline to be overcome by a more sophisticated classifier.

The results for the first experiment are described in Table 2.

**Table 2.** Accuracy for each algorithm for the first experiment.

Features	<i>Naïve Bayes</i>	<i>C4.5</i>	<i>SVM</i>
1 <i>sentences</i>	59.00 %	57.84 %	63.68 %
2 <i>verb inflections (inf)</i>	57.56 %	58.84 %	60.64 %
3 <i>+ 6-word context window (inf-6cw)</i>	59.12 %	60.16 %	61.32 %
4 <i>+ adverbs, prepositions and conjunctions (inf-6cw-adv-prep-cj)</i>	68.00 %	64.88 %	72.84 %

The results illustrated in the Table were obtained using the same training and testing datasets, so that several features and algorithms could be tested. Taking into account the goal of this task, the most straightforward approach would be to use the complete sentences – ARG<sub>1</sub> and ARG<sub>2</sub> – as features (feature#1). The first assumption when using this feature was that it might contain too much noise, as the complete sentences were being used. However, the sentences might also contain some singularities that help the classifier to achieve results close to the baseline. Feature#2 comprises the inflections of all the verbs in both sentences. By expressing an event, the verb can be a relevant source of information when regarding discourse relations and their specific inflections could help to identify the presence of a discourse relation or not. Despite empirically it could be a very relevant feature, by itself this feature achieves results below the baseline when considering all the algorithms tested.

As suggested by Wellner et al. (2006), we finally essayed to combine several features [20]. When combining the verb inflections with the 6-word context window – composed by three words in the end of ARG<sub>1</sub> and three words in the beginning of ARG<sub>2</sub> (feature#3) – we were able to improve the accuracy of all



the three classifiers. With this configuration, it is even possible to overcome the baseline with *SVM*. Finally, feature#4 includes the combination of the previous features with all the adverbs, prepositions and conjunctions found in both arguments (results in the fourth line of the Table). Using this combination, we were able to significantly improve the results of the three classifiers, with all results scoring above the baseline.

Yet, when analyzing the behavior of each classifier, we can conclude that for all of them the combination of the features keeps enhancing their accuracy. Despite this, the results obtained using *SVM* are still the best ones, being more than 10 percentage points above the baseline.

After finding a combination of features that overcomes the baseline with all the algorithms, a new experiment was performed, varying only the size of the training dataset. Thus, the feature used was always the same (*inf-6cw-adv-prep-cj*), so as the algorithms and the testing dataset.

The second experiment aims then to verify if extending the training dataset would improve the accuracy of the classifiers. Table 3 reports the results for the training dataset extensions.

**Table 3.** Accuracy when extending the for each algorithm.

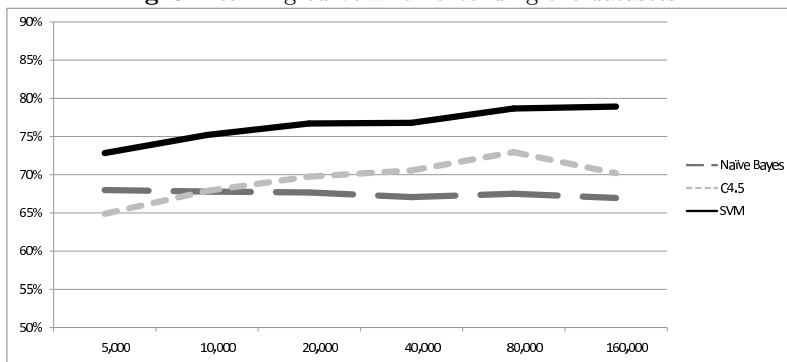
Number of pairs	<i>Naïve Bayes</i>	<i>C4.5</i>	<i>SVM</i>
<i>5,000 pairs</i>	68.00 %	64.88 %	72.84 %
<i>10,000 pairs</i>	67.80 %	67.88 %	75.20 %
<i>20,000 pairs</i>	67.68 %	69.76 %	76.72 %
<i>40,000 pairs</i>	67.08 %	70.56 %	76.80 %
<i>80,000 pairs</i>	67.52 %	72.96 %	78.68 %
<i>160,000 pairs</i>	66.96 %	70.20 %	78.92 %

The first line was obtained by training all the algorithms using a dataset containing 5,000 pairs. These are the final values reported in the previous experiment. The training dataset was then duplicated until the learning curve has reached a point where no relevant improvements were obtained. The learning curve is illustrated in Figure 3.

Note that all the training algorithms keep performing better when doubling the training dataset until the 80,000 pairs mark. At this point, there are slight improvements (case of *SVM*) or even worse performances (cases of *Naïve Bayes* and *C4.5*). In conclusion, the best performing algorithm – *SVM* – was used in a training dataset of 160,000 pairs to perform the first step of the connective insertion procedure: identify if two sentences enter a discourse relation or not.

*Relations* vs. *Relations*. Once the previous classifier has determined that the sentences share a discourse relation, it is now time to identify which relation is that.

**Fig. 3.** Learning curve when extending the datasets.



This is a multi-class classification problem, as we have 18 possible discourse relations to assign. Recall Figure 2 that shows the distribution of the classes in the testing dataset. Note that the most frequent discourse relation class is CONTINGENCY-CAUSE-RESULT. A baseline for this classification problem would assign the most frequent class to all the instances in the dataset, so that it would achieve an accuracy of 27%, being this value the lower boundary to be overcome by a more sophisticated classifier.

The first experiment takes together all the 18 classes in a all-vs-all approach. A training dataset containing 2,500 split unevenly through all the classes was used. In the same way, the testing dataset contains 2,500 pairs aiming to reflect the normal distribution of the discourse relations in a corpus. Results for this experiment are reported in Table 4.

**Table 4.** Accuracy for all the classes using a all-vs-all approach.

Classes	<i>Naïve Bayes</i>	<i>C4.5</i>	<i>SVM</i>
ALL CLASSES	22.64 %	23.36 %	29.6 %

As these results point out, deciding between 18 different classes at the same time is a very hard task. Even though the best result (*SVM*) is slightly above the baseline, this is a very poor accuracy.

Hence, this problem was split into several problems, assuming a one-vs-all approach. Thus, for each class, we trained a classifier that aims to determine if a given pair of arguments share that relation or any of the other relation. This way, we turned a multi-class problem into a binary classification problem.

Taking into account that *SVM* had the best performance in the previous experiment, and that it is specially suitable for binary classification, this was the algorithm used in this experiment. Also the same combined features (*inf-6cw-adv-prep-cj*) – found in the first experiment – were used to train the clas-

sifiers. This experiment is based on training datasets containing 2,500 pairs divided in two: 1,250 from the specific relation and 1,250 from all the others. The goal was to build the training datasets with the same number of instances of the classes. However, we were unable to obtain in the corpus 1,250 for three classes (CONTINGENCY-CONDITION-FACTUAL, EXPANSION-ALTERNATIVE-DISJUNCTIVE and CONTINGENCY-CONDITION-CONTRA-FACTUAL). For each of these classes we built the training dataset by including the maximum number of instances for each class (66, 458 and 927, respectively) and the same number of instances of all the other classes. Thus, the training datasets for these three classes contained a total of 132, 916 and 1854, respectively.

Table 5 details the accuracy values obtained when training a single classifier for each class, using training datasets containing 2,500 pairs.

**Table 5.** Accuracy for each class using a one-vs-all approach.

Classes	<i>SVM</i>
CONTINGENCY-CONDITION-FACTUAL	61.43 %
EXPANSION-ALTERNATIVE-DISJUNCTIVE	65.38 %
CONTINGENCY-CONDITION-CONTRA-FACTUAL	80.83 %
COMPARISON-CONCESSION-CONTRA-EXPECTATION	89.26 %
EXPANSION-EXCEPTION	74.13 %
EXPANSION-ALTERNATIVE-CHOSEN-ALTERNATIVE	64.87 %
EXPANSION-RESTATEMENT-GENERALIZATION	70.53 %
TEMPORAL-SYNCHRONOUS	62.23 %
TEMPORAL-ASYNCHRONOUS-PRECEDENCE	89.74 %
CONTINGENCY-CONDITION-HYPOTHETICAL	77.40 %
COMPARISON-CONCESSION-EXPECTATION	75.48 %
EXPANSION-RESTATEMENT-EQUIVALENCE	71.89 %
TEMPORAL-ASYNCHRONOUS-SUCCESSION	76.60 %
CONTINGENCY-CAUSE-REASON	60.51 %
EXPANSION-INSTANTIATION	75.80 %
EXPANSION-ADDITION	89.34 %
COMPARISON-CONTRAST-OPPOSITION	69.18 %
CONTINGENCY-CAUSE-RESULT	62.83 %

The results shown in the Table point out that all the classifiers performed significantly better when compared with the first experiment where the classes were considered altogether. Moreover, by using a one-vs-all approach we were able to create classifiers for each discourse class which are highly above the baseline and which are able to distinguish a specific class from all the other possible classes.

### 3 Final remarks

This paper presents an approach to find discourse relations between sentences, in order to select a discourse connective to be inserted between those sentences. The procedure uses a sequence of classifiers, firstly to determine if there is any relation between the two sentences, and, afterwards, to distinguish which relation is that. By uncovering the discourse relation and selecting the corresponding connective, this work seeks to go a step forward in improving the quality of a text. The accuracy results of all the classifiers are very promising, suggesting that the probability of finding the correct connective is on average 72%.

The textual quality of a summary (e.g. fluency, readability, discourse coherence, etc.) has been repeatedly reported as the main flaw in current automatic summarization technology. Considering this, the ultimate goal of the procedure presented in this paper is to be included in a post-processing module of an automatic multi-document summarization system, that creates summaries using extraction methods. Post-processing is a module composed by three tasks executed in sequence – sentence reduction, paragraph creation and connective insertion. While sentence reduction aims to remove extraneous information from the summary, paragraph creation seeks to define topics of interest in the text. Yet, connective insertion is applied over the sentences in each paragraph by inserting between them the appropriate discourse connective (if any), creating interconnected text. Thus, the motivation behind this work is to seek for improvements in respect to the final quality of a summary built using extractive methods.

### References

1. Biran, O., Rambow, O.: Identifying justifications in written dialogs by classifying text as argumentative. *Int. J. Semantic Computing* 5(4), 363–381 (2011)
2. Blair-Goldensohn, S., McKeown, K., Rambow, O.: Building and refining rhetorical-semantic relation models. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. pp. 428–435. Association for Computational Linguistics, Rochester, New York (April 2007)
3. Feng, V.W., Hirst, G.: Text-level discourse parsing with rich linguistic features. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers – Volume 1*. pp. 60–68. ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
4. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. pp. 338–345. UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
5. Lapata, M., Lascarides, A.: Inferring sentence-internal temporal relations. In: *HLT-NAACL*. pp. 153–160 (2004)
6. Lee, A., Prasad, R., Joshi, A., Dinesh, N.: Complexity of dependencies in discourse: Are dependencies in discourse more complex than in syntax? p. 12. *Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories.*, Prague, Czech Republic (December 2006)

7. Lin, Z., Kan, M.Y., Ng, H.T.: Recognizing implicit discourse relations in the Penn Discourse Treebank. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1. pp. 343–351. EMNLP 2009: Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
8. Louis, A., Joshi, A., Nenkova, A.: Discourse indicators for content selection in summarization. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 147–156. SIGDIAL '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
9. Marcu, D., Echihiabi, A.: An unsupervised approach to recognizing discourse relations. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 368–375. ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002)
10. Park, J., Cardie, C.: Improving implicit discourse relation recognition through feature set optimization. In: Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 108–112. SIGDIAL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
11. Pitler, E., Nenkova, A.: Using syntax to disambiguate explicit discourse connectives in text. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. pp. 13–16. ACLShort '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
12. Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., Joshi, A.: Easily identifiable discourse relations. In: Coling 2008: Companion volume: Posters. pp. 87–90. Coling 2008 Organizing Committee, Manchester, UK (August 2008)
13. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse TreeBank 2.0. In: Proceedings of LREC (2008)
14. Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., Webber, B.: The Penn Discourse Treebank 2.0 annotation manual. Tech. Rep. IRCS-08-01, Institute for Research in Cognitive Science, University of Pennsylvania (Dec 2007)
15. Quinlan, J.R.: Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research* 4(1), 77–90 (Mar 1996)
16. Rocha, P., Santos, D.: CETEMPúblico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. In: 5th. pp. 131–140 (2000)
17. Silveira, S.B., Branco, A.: Combining a double clustering approach with sentence simplification to produce highly informative multi-document summaries. In: IRI 2012: 14th International Conference on Artificial Intelligence. pp. 482–489. Las Vegas, USA (August 2012)
18. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)
19. Versley, Y.: Subgraph-based classification of explicit and implicit discourse relations. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers. pp. 264–275. Association for Computational Linguistics, Potsdam, Germany (March 2013)
20. Wellner, B., Pustejovsky, J., Havasi, C., Rumshisky, A., Saurí, R.: Classification of discourse coherence relations: an exploratory study using multiple knowledge sources. In: Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue. pp. 117–125. SigDIAL '06, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)
21. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (2005).