Semi-supervised Discourse Relation Classification with Structural Learning

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka

Graduate School of Information Science & Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
hugo@mi.ci.i.u-tokyo.ac.jp, danushka@iba.t.u-tokyo.ac.jp,
ishizuka@i.u-tokyo.ac.jp

Abstract. The corpora available for training discourse relation classifiers are annotated using a general set of discourse relations. However, for certain applications, custom discourse relations are required. Creating a new annotated corpus with a new relation taxonomy is a time-consuming and costly process. We address this problem by proposing a semi-supervised approach to discourse relation classification based on Structural Learning. First, we solve a set of auxiliary classification problems using unlabeled data. Second, the learned classifiers are used to extend feature vectors to train a discourse relation classifier. By defining a relevant set of auxiliary classification problems, we show that the proposed method brings improvement of at least 50% in accuracy and F-score on the RST Discourse Treebank and Penn Discourse Treebank, when small training sets of ca. 1000 training instances are employed. This is an attractive perspective for training discourse relation classifiers on domains where little amount of labeled training data is available.

1 Introduction

Detecting the discourse relations underlying the different units of a text is crucial for several NLP applications, such as text summarization [1] or dialogue generation [2]. To date, only three major annotated corpora are available, the RST Discourse Treebank (RSTDT) [3], the Discourse Graphbank [4], and the Penn Discourse Treebank (PDTB) [5]. The RSTDT is based on the Rhetorical Structure Theory framework (RST) [6], and annotation is done using a set of 78 fine-grained discourse relations, usually grouped by researchers into a set of 18 more general relations [3]. In the Discourse GraphBank, annotation is done using a set of 11 discourse relations. Finally, in the PDTB, annotation is done in a hierarchical fashion, with 4 relations at the highest-level, and 20 at the most detailed level.

However, in some applications, we must extract discourse relations that are different from the ones defined in above-mentioned discourse theories. In [7] for instance, it is shown that the use of a RST discourse parser improves the detection of relevant information in clinical guidelines. Notably, certain RST discourse relations such as Temporal or Consequence are useful in the context

A. Gelbukh (Ed.): CICLing 2011, Part I, LNCS 6608, pp. 340–352, 2011.

of clinical information extraction. However, the majority of RST relations are too generic and not relevant enough for this task. For this application, capturing relations such as PERMISSION, OBLIGATION or ADVICE would be of greater interest. Thus, in the context of a specialized application, employing a discourse relation classifier trained on a custom set of discourse relations can be required.

A straightforward solution consists of creating a new corpus annotated with the desired set of discourse relations. However, this process is costly and time-consuming. Alternatively, it is interesting to tackle the discourse relation classification problem by employing a semi-supervised approach. While having at our disposition a small set of labeled examples, we propose to leverage freely-available unlabeled data, by employing Structural Learning [8]. In the proposed approach, unlabeled training data is employed to solve auxiliary classification tasks related to the main discourse classification problem. The unlabeled data can be obtained with a minimal effort, for instance on the web. By solving the auxiliary tasks, some information about the main discourse classification task is learnt, and encoded as new features in the main classifier's training and test feature vectors. We show that the proposed method brings a significant improvement in classification performance (F-score and accuracy), in particular when training sets of small to moderate (ca. 100 to 1000 instances) size are employed.

Our contributions in this paper are summarized as follows.

- We propose a set of auxiliary tasks related to the main discourse relation classification problem, and which can be solved using unlabeled data only. We show that incorporating these tasks into the main problem through Structural Learning brings significant improvement in F-score and classification accuracy of at least 50%, when small to moderate amounts of training data are used.
- The proposed method is evaluated on the RSTDT and PDTB corpus, and compared to a state-of-the-art semi-supervised discourse relation classification method [9].

2 Related Work

Most of the recent work on discourse relation classification have been based on either fully-supervised or unsupervised methods.

The first unsupervised approach to discourse relation classification was presented in [10]. In this work, the authors were the first to employ word pair features calculated from the two arguments of a relation. These features have the promise of capturing *implicit relations*, i.e. discourse relations not signaled by a discourse cue, such as *but*, *and* or *thus*. For instance, the presence of a word pair (*flashy, low-key*) indicates a CONTRAST relation.

Supervised methods have been employed to train discourse relation classifiers on the RSTDT. In [11], as a part of the sentence-level discourse sparser 'SPADE', a probabilistic model employing lexical and syntactic features is used for training a discourse relation classifier. In [12], for the same task, relation classification is done using a Support Vector Machines [13]-based classifier trained on a rich

set of shallow lexical and syntactic features. In recent work [14], another RST parser based on a chart-parsing approach is presented. Here, discourse relation classification is performed using a neural network trained on syntactic and lexical features, including lexical heads.

Supervised methods have also been employed to learn discourse relation classifiers on the PDTB. In [15], an explicit discourse relation classifier is presented. Explicit relations are discourse relations signaled by a discourse cue, and the authors demonstrate that these can be classified accurately, with an F-score of 0.93. However, implicit discourse relations have been shown to be much more difficult to classify. In [16], implicit discourse relation classification on the PDTB is studied. The authors employ features such as word pairs, verb classes, modality, context, lexical features, and obtain a state-of-the-art accuracy of 0.446. In [17], for the same task, the authors also employ word pairs, as well as dependency paths, contextual information, and production rules in parse trees. They obtain an accuracy of 0.402.

Semi-supervised learning methods have been employed for a variety of tasks in NLP, such as named-entity recognition or text classification. In particular, [8] have presented the Structural Learning theory, which is based on the prediction of properties of the main classification task, using unlabeled data only. Their algorithm has been shown to perform at least as well as co-training [18] for several tasks. Structural Learning is conceptually similar to Multi-task learning [19], where related problems are learnt at the same time as the main classification problem, which enables inductive transfer and leads to a better model.

To the best of our knowledge, our previous work [9] corresponds to the first semi-supervised discourse relation classification method. In that work, a method based on the co-occurrence of features observed in unlabeled data was introduced. The degree of co-occurrence between feature pairs is first measured on a set of sentences extracted from Wikipedia[1], using the χ^2-measure [20]. Co-occurrence information is then used to extend the feature vectors of a discourse relation classifier, bringing additional information about features unseen during training. The feature set contains word pairs computed between the arguments of discourse relations, production rules from the parse trees, as well as lexical heads. This co-occurrence-based method brings significant increase in classification performance when training is done on small sets, containing few instances of certain discourse relations.

In this paper, we employ a feature set similar to [9], but propose a different semi-supervised learning method to tackle the discourse relation classification task. First, whereas the co-occurrence-based method employs unlabeled data to learn feature co-occurrences, the proposed method uses unlabeled data to solve auxiliary classification problems. Second, the co-occurrence-based method encodes co-occurrence information into a large set of new features, which is then appended to the original feature vectors. Because the size of the appended feature set depends on the number of unseen features during training, for small training sets, which correspond to a large number of unseen features, the process results

[1] `http://en.wikipedia.org`

in a considerable dimension increase, typically of ca. 10000. By contrast, in the proposed method, the information learned from unlabeled data is encoded into a compact set of new features, typically less than 100, and including these features into the classification problem does not increase dimension considerably. Nonetheless, because our experimental setting is similar, the proposed method can be directly evaluated against the feature co-occurrence-based discourse relation classification method introduced in [9].

3 Method

In this paper, we aim at learning a discourse relation classifier, given a set of labeled instances T and a set of unlabeled instances L, where typically $|L| >> |T|$. The main idea is to use the unlabeled instances to generate auxiliary tasks that are useful for discovering important properties about the structure of the main problem. If the auxiliary tasks are similar—or at least related—to the main discourse relation classification task, then we will benefit from solving them. For instance, consider the following REASON relation, holding between two discourse units in square brackets.

REASON: [Our research shows we sell more of our heavier issues] [because readers believe they are getting more for what they pay for.]

In discourse relation classification, it occurs often that a discourse relation can be predicted by observing the word pairs of its arguments. For instance, trivially, a word pair ($*$, *but*) is usually the indication of a CONTRAST relation. In our example, the word pair (*show*, *because*), which has been lemmatized to be made more general, is a strong indicator of the REASON relation. Intuitively, a task related to detecting the REASON relation will thus be the task of detecting the (*show*, *because*) word pair, when observing other features of the instance. A positive training instance of this new auxiliary task would be,

+(*show*, *because*): [Our research ____ we sell more of our heavier issues] [____ readers believe they are getting more for what they pay for.]

The original word pair has to be masked in order to make for an acceptable training instance. A negative training instance for this auxiliary task would be any instance not containing the word pair (*show*, *because*), such as,

−(*show*, *because*): [She has thrown extravagant soirees for crowds of people,] [but prefers more intimate gatherings.]

This auxiliary task, which is related to the task of predicting the REASON relation, can be learned using unlabeled data only. In Section 3.1, we detail the Structural Learning algorithm, and show how the information learned from solving these auxiliary tasks can be included into the main classification problem. Then, we present the features employed for this task in Section 3.2. Finally, we describe the auxiliary problem creation step in Section 3.3.

3.1 Structural Learning

In this section, we assume that we have at our disposition a training set consisting of T labeled instances $\left\{(\boldsymbol{x_t}, y_t)_{t=1}^{T}\right\}$, where the $\boldsymbol{x_t}$ are feature vectors and y_t the class labels. The feature space has dimension d.

First, we solve a set of auxiliary classification problems p_l for $l \in [1, \ldots, m]$, using linear classifiers, and find for each p_l the optimal weight vector $\hat{\boldsymbol{w_l}}$ such that,

$$\hat{\boldsymbol{w_l}} = \arg\min_{\boldsymbol{w}} \left(\sum_j L(\boldsymbol{w} \cdot x_j, p_l(x_j)) + \lambda ||\boldsymbol{w}||^2 \right). \tag{1}$$

Here L is a loss function and λ a regularization coefficient.

Next, we stack the optimal weight vectors of each problem column-per-column, and create a matrix $W = [\hat{\boldsymbol{w_1}} \ldots \hat{\boldsymbol{w_m}}]$. In order to reduce the dimension of W, we perform on this matrix a singular value decomposition (SVD). It is noteworthy that whereas algorithms such as principal component analysis aim at reducing the dimension of the feature space, performing a SVD on W is a dimension reduction on the space of auxiliary classifiers, aimed at learning a compact representation of it.

Since typically discourse relation classifiers employ several types of heterogenous features, such as words, part-of-speech tags and word pairs, it is reasonable to perform a localized dimension reduction for each type of feature. Consequently, we perform a series of 'block' SVDs, for each type of feature employed. For each feature type f_i, $i \in [1, \ldots, n]$, whose index in the feature space starts at position s_i, and ends at position e_i, we create a feature type-specific structural parameter matrix θ_i so that,

$$U_i, D_i, V_i^T = \text{SVD}(W_{[s_i:e_i,:]}) \tag{2}$$

$$\theta_i = U_{i[1:h,:]}^T \tag{3}$$

The number h is the number of structural features we wish to incorporate in our problem.

The complete structural parameter matrix $\theta = [\theta_1 \ldots \theta_n]$ has dimension $h \times d$, and it encodes the structure learnt by the auxiliary tasks in a low-dimension common space. We can now project each training and test feature vector of the main task on θ, and obtain a set of h new structural features, which are appended to their original feature vector. We obtain the training set,

$$\left\{ \left(\begin{bmatrix} \boldsymbol{x_t} \\ \theta \boldsymbol{x_t} \end{bmatrix}, y_t \right)_{t=1}^{T} \right\} \tag{4}$$

Finally, we rescale the extended features. As observed in [21], we found necessary to give relatively more weight to the structural features, which can be

performed by rescaling them. This is done by finding a factor $k \in \mathbb{R}^+, k > 1$ so that,

$$\sum_{t=1}^{T} ||\theta \boldsymbol{x_t}|| = k \sum_{t=1}^{T} ||\boldsymbol{x_t}||. \quad (5)$$

This factor is found empirically, as the value that maximizes classification accuracy on a held-out dataset.

3.2 Features

We use three types of features, which have previously been employed successfully in several works presented in Section 2, including our co-occurrence-based semi-supervised method [9]: Word pairs, production rules from the parse tree, as well as features encoding the lexico-syntactic context at the border between two units of text [11]. Word pairs are lemmatized using the Wordnet-based lemmatizer of NLTK [22].

Figure 1 shows the parse tree for a sentence composed of two discourse units, which serve as arguments of a discourse relation we want to generate a feature vector from. Lexical heads have been calculated using the projection rules of [23], and annotated between brackets. Surrounded by dots is, for each argument, the minimal set of sub-parse trees containing strictly all the words of the argument.

We first extract all possible lemmatized word-pairs from the two arguments, such as (*Mr.*, *when*), (*decline*, *ask*) or (*comment*, *sale*). Next, we extract from left and right argument separately, all production rules from the sub-parse trees, such as NP $\mapsto$ NNP NNP, NNP $\mapsto$ "Sherry" or TO $\mapsto$ "to".

Finally, we encode in our features three nodes of the parse tree, which capture the local context at the connection point between the two arguments: The first node, which we call N_w, is the highest ancestor of the first argument's last word w, and is such that N_w's right-sibling is the ancestor of the second argument's first word. N_w's right-sibling node is called N_r. Finally, we call N_p the parent of N_w and N_r. For each node, we encode in the feature vector its part-of-speech (POS) and lexical head. For instance, in Figure 1, we have N_w = S(comment), N_r = SBAR(when), and N_p = VP(declined). In the PDTB, certain discourse relations have disjoint arguments. In this case, as well as in the case where the two arguments belong to different sentences, the nodes N_w, N_r, N_p cannot be defined, and their corresponding features are given the value zero.

3.3 Auxiliary Classification Problems

Following the intuition presented at the beginning of this section, we use as auxiliary tasks the prediction of word pairs in unlabeled data, when observing all other features. The creation of training data for the auxiliary task of predicting the presence of word pair (w_1, w_2) is done as follows:

1. We filter out unlabeled instances containing the word pair (w_1, w_2). These will serve as positive training examples for the auxiliary task.

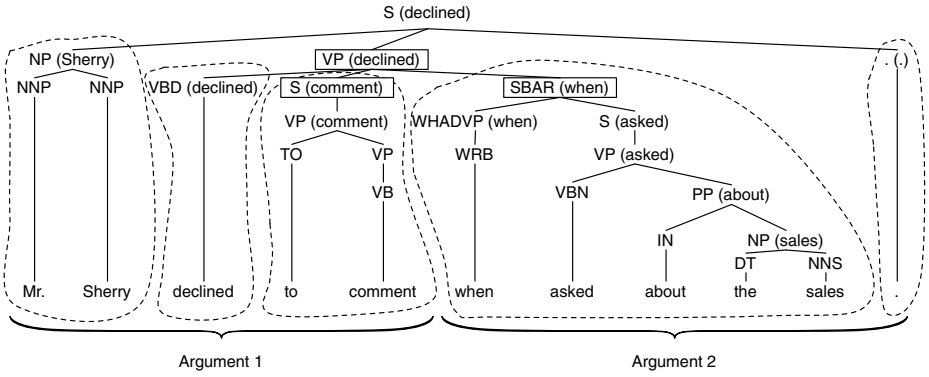


Fig. 1. Two arguments of a discourse relation, and the minimum set of subtrees that contain them—lexical heads are indicated between brackets

2. The remaining unlabeled instances, i.e. those which do not contain the word pair (w_1, w_2), will serve as negative training examples.
3. Since typically there are many more negative training examples than positive ones, there is a risk that the classifier might label every new test instance as belonging to the negative class. To prevent this issue, we cap the number of negative training examples. We empirically found that using a $2:1$ ratio of negative to positive training instances gave the best results. Using a lower ratio gave slightly worse results, while using higher ratios of negative training data did not significantly change the performance, but increased the training time of the auxiliary classifiers.
4. In positive and negative training instances, all word pair features are masked (set to zero). Although we could choose to keep certain word pairs unmasked, [8] recommend for optimal performance to mask (and predict) all features that have a good correlation to the labels of the auxiliary tasks (the other word pairs).

4 Experiments

In [8], it is shown that setting the number of structural features h between 20 and 100 does not change the results significantly. We select the intermediate value $h = 50$, which is used in all the following experiments. The factor used to rescale structural features is empirically set to five, which is consistent with the results of [21].

We employ as our unlabeled data the set of $100,000$ unlabeled instances used in [9], which consists of sentences randomly extracted from Wikipedia and segmented into elementary discourse units automatically. The sentences have been parsed using the Stanford parser [24], in order to extract syntactic information. With $100,000$ unlabeled training instances, it occurs often that the auxiliary classification task corresponding to the detection of a word pair will have very

few positive training examples—typically around ten. To avoid incorporating into the structural features auxiliary problems whose classification performance is poor, we filter out auxiliary problems with less than 30 positive training instances. This finally results in solving 1358 auxiliary problems for RSTDT relation classification, and 1542 for PDTB.

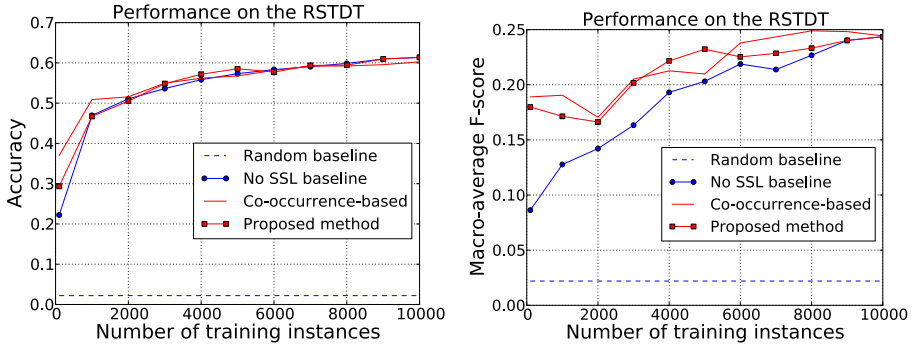
We follow the common practice in discourse research for partitioning the discourse corpora into training and test set. For the RST classifier, the dedicated training and test sets of the RSTDT are employed. For the PDTB classifier, we conform to the guidelines of [25, 5]: The portion of the corpus corresponding to sections 2–21 of the WSJ is used for training the classifier, while the portion corresponding to WSJ section 23 is used for testing. This setting is identical to the one employed in [9].

For RSTDT, we extract 25078 training vectors and 1633 test vectors. For PDTB we extract 49748 training vectors and 1688 test vectors. There are 41 classes (relation types) in the RSTDT relation classification task, and 29 classes in the PDTB task. For the PDTB, we select level-two relations, because they have better expressivity and are not too fine-grained. For our classifiers, we use the multi-class logistic regression (maximum entropy model) implemented in the Classias toolkit [27]. Regularization parameters are set to their default value of one and are fixed throughout the experiments described in the paper.

In the following experiments, we evaluate the performance of the proposed method against two baselines. The first is the 'random' baseline, in which classification decisions are made randomly. The second, noted *no SSL* in Figures 2 and 3, is the classifier trained with the same feature set, on the same training set as the proposed method, but for which no semi-supervised learning algorithm has been applied. As in [9], we employ macro-average F-score as the proposed evaluation metric. Indeed, since training sets can be imbalanced due to the prevalence of certain well-detected relations, such as ELABORATION or ATTRIBUTION in the case of the RSTDT, the micro-average F-score does not reflect accurately the classifier's performance on all classes. The macro-average F-score, which is the arithmetic mean of the F-score computed for each class, considers each class with equal importance.

We first measure the performance on the RSTDT when 100 to 10000 training instances are used. For each training set size, all classifiers are trained with the same instances. Results are indicated in Figure 2. We observe that the proposed method improves accuracy compared to the *no SSL* baseline only for 100 training instances. For both the proposed method and the co-occurrence-based method [9], above 2000 training instances, accuracy scores are as high as the *no SSL* baseline. However, we see a clear performance improvement over *no SSL* in terms of macro-average F-score. For 100 training instances, this baseline classifier has a macro-average F-score of 0.086. The classifier trained with the proposed method reaches a macro-average F-score of 0.180 (+108.34% score increase over the *no SSL* baseline), while the co-occurrence-based classifier obtains an F-score of 0.189 (+119% increase over *no SSL*). For 1000 training instances, the *no SSL* baseline has an F-score of 0.127, while the classifier trained with the proposed method

reaches an F-score of 0.171 (+34.38% over *no SSL*). The co-occurrence-based classifier obtains a slightly higher F-score of 0.191 (+49.18% over *no SSL*). From 1000 to 9000 training instances, we observe in each case an F-score increase over *no SSL*, although the relative performance gain diminishes gradually. Finally, when 10000 training instances are used, both semi-supervised methods obtain the same F-score as *no SSL*, at around 0.244. As in the case of co-occurrence-based discourse relation classification [9], we observe that the proposed method is most efficient when small training sets are employed, whereas there is no performance gain when using larger sets of 10000 training instances.

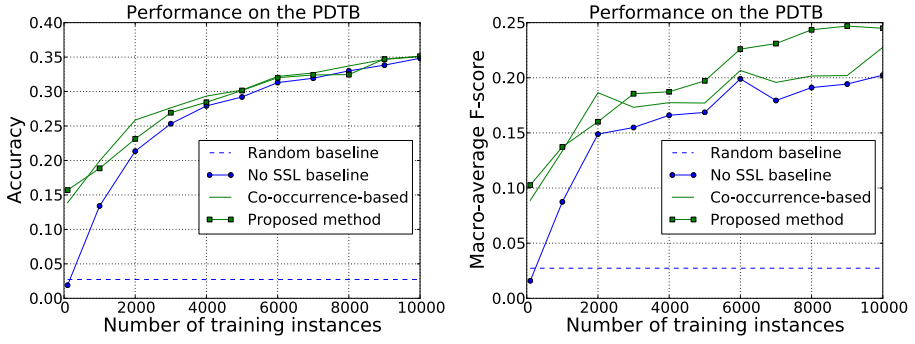


(a) Accuracy (b) Macro-average F-score

Fig. 2. Scores on the RSTDT, as a function of the number of training instances used

Similarly, we measure the performance of the proposed method on the PDTB. The results of this experiment are indicated in Figure 3. We observe a similar trend as in the case of the RSTDT experiments. For 100 training instances, the *no SSL* baseline has an extremely low accuracy of 0.019 and a macro-average F-score of 0.016. However, the classifier trained with the proposed method reaches a respective accuracy and F-score of 0.157 (+726.84% score change over the *no SSL* baseline) and 0.103 (+545.91% over *no SSL*). These scores are slightly higher than the co-occurrence-based classifier, which reaches respective accuracy and F-score of 0.139 (+630% over *no SSL*) and 0.089 (+459.12% over *no SSL*). When 1000 training instances are employed, using semi-supervised methods results in a clear improvement both in accuracy and F-score. The *no SSL* baseline obtains an accuracy of 0.134 and F-score of 0.087, while the proposed method reaches a respective accuracy and F-score of 0.189 (+40.75% over *no SSL*) and 0.137 (+56.91% over *no SSL*). In this case, the co-occurrence-based method obtains a respective accuracy and F-score of 0.199 (+48.73% over *no SSL*) and 0.134 (+52.69% over *no SSL*). On this dataset, the proposed method outperforms the co-occurrence-based method when more than 2000 training instances are used. Notably, for 9000 training instances, whereas *no SSL*'s macro-average F-score is 0.194, the proposed method reaches an F-score of 0.247 (+27.07% over *no SSL*), versus 0.202 (+3.96% over *no SSL*) for the co-occurrence-based classifier.

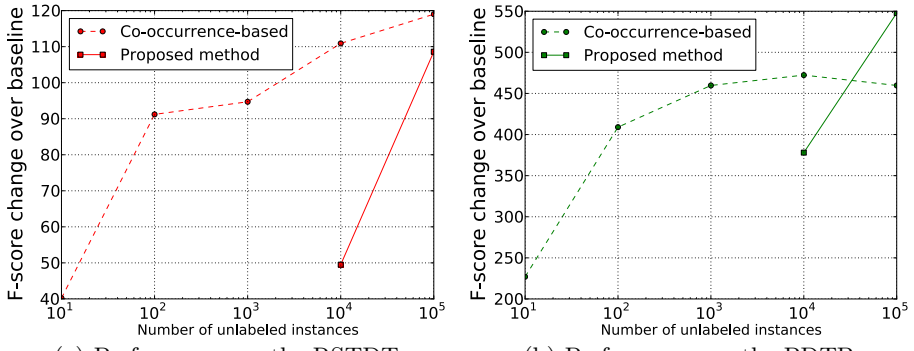
These scores are consistent with the results of Figure 2, with the exception that, for PDTB relation classification, the proposed method did improve the macro-average F-score when large training sets were used.



(a) Accuracy (b) Macro-average F-score

Fig. 3. Scores on the PDTB, as a function of the number of training instances used

An interesting property of a semi-supervised method is how its performance will be affected by the amount of unlabeled training data employed. After selecting a training set of 100 instances, we evaluate the performance of the proposed method when a variable amount of unlabeled training data is used. The results are shown in Figure 4. A first observation is that, when only 10, 100 or 1000 unlabeled instances are available, there is not enough data to train the auxiliary classifiers, and consequently the proposed method cannot be applied. On the other hand, the co-occurrence-based method performs well even with small amounts of unlabeled training data: With 10 unlabeled instances, this method increases the F-score for RSTDT by 40.2%, and by 227% for PDTB. For 10000 unlabeled training instances, it becomes possible to train the auxiliary classifiers. In this case, the proposed method scores lower than the feature co-occurrence-based method, with an F-score increase on the RSTDT of 110.9% for the co-occurrence-based method, vs. 49.5% for the proposed method. For the PDTB the F-score increase is 472.3% for the co-occurrence-based method, against 378% for the proposed method. Finally, when using the full set of 100,000 unlabeled training instances, the performance of the proposed method increases dramatically, and becomes very close to the performance of the co-occurrence-based method. For RSTDT relation classification, we observe an F-score increase of 119.0% for the co-occurrence-based method, against 108.6% for the proposed method. However, for PDTB relation classification, the proposed method outperforms the co-occurrence-based method, with an F-score increase of 547.8% for the proposed method, against 459.7% for the co-occurrence-based method. These values confirm that, provided that we have a sufficient amount of unlabeled training data at our disposition, the proposed method performs at least as well as the co-occurrence-based discourse relation classification method [9].



(a) Performance on the RSTDT (b) Performance on the PDTB

Fig. 4. Effect of unlabeled data on the proposed method, for 100 training instances

Finally, we discuss some qualitative differences between the proposed method and the co-occurrence-based method [9]. First, whereas the co-occurrence-based method performs a large increase in the size of the feature space—dimension increase of ca. 15000 for a training set of 100 instances—the proposed method only adds a small, fixed number of features—set to 50 in our experiments. Then, the proposed method was shown to require more unlabeled data than the co-occurrence-based method, in order to train the auxiliary classification problems. Indeed, with 100 or 1000 unlabeled instances, most word pairs rarely occur in unlabeled data, which makes it impossible to train accurate auxiliary classifiers. Last, whereas the feature co-occurrence based method is independent from any classification problem or machine learning algorithm, the proposed method requires some human supervision in order to define relevant auxiliary tasks, and it requires employing linear classifiers to solve the auxiliary classification problems.

5 Conclusion

We presented a semi-supervised discourse relation classification method based on Structural Learning [8]. The method was evaluated on the RSTDT and PDTB, where it was shown to bring significant performance increase in accuracy and F-score, especially in the cases where small training sets of ca. 1000 instances were used. This is an interesting outlook for creating discourse relation classifiers on domains with little available training data.

The proposed method was compared to a feature co-occurrence-based method [9], and it was shown to perform comparably given the same amount of unlabeled data. Although the relative performance improvement over baseline classifiers is important, classification accuracy and macro-average F-score are rather low when large training sets are employed. We hypothesize that this is due to the poor detection of implicit relations, where the current state-of-the-art F-score is still modest. However, ongoing research has been focusing on finding appropriate features for this task [28,29], which has the promise of enabling us to improve classification performance.

References

1. Louis, A., Joshi, A., Nenkova, A.: Discourse indicators for content selection in summarization. In: Proc. of SIGDIAL 2010, pp. 147–156 (2010)
2. Piwek, P., Hernault, H., Prendinger, H., Ishizuka, M.: Generating dialogues between virtual agents automatically from text. In: Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 161–174. Springer, Heidelberg (2007)
3. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In: Proc. of Second SIGdial Workshop on Discourse and Dialogue, vol. 16, pp. 1–10 (2001)
4. Wolf, F., Gibson, E.: Representing discourse coherence: A corpus-based study. Computational Linguistics 31, 249–287 (2005)
5. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse TreeBank 2.0. In: Proc. of LREC 2008 (2008)
6. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: Toward a functional theory of text organization. Text 8, 243–281 (1988)
7. Georg, G., Hernault, H., Cavazza, M., Prendinger, H., Ishizuka, M.: From rhetorical structures to document structure: Shallow pragmatic analysis for document engineering. In: Proc. of DocEng 2009, pp. 185–192. ACM, New York (2009)
8. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. J. Mach. Learn. Res. 6, 1817–1853 (2005)
9. Hernault, H., Bollegala, D., Ishizuka, M.: A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In: Proc. of EMNLP 2010, pp. 399–409 (2010)
10. Marcu, D., Echihabi, A.: An unsupervised approach to recognizing discourse relations. In: Proc. of ACL 2002, pp. 368–375 (2002)
11. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Proc. of NA-ACL 2003, vol. 1, pp. 149–156 (2003)
12. duVerle, D.A., Prendinger, H.: A novel discourse parser based on Support Vector Machine classification. In: Proc. of ACL 2009, pp. 665–673 (2009)
13. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
14. Sagae, K.: Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In: Proc. of IWPT 2009, pp. 81–84 (2009)
15. Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., Joshi, A.: Easily identifiable discourse relations. In: Proc. of COLING 2008 (Posters), pp. 87–90 (2008)
16. Pitler, E., Louis, A., Nenkova, A.: Automatic sense prediction for implicit discourse relations in text. In: Proc. of ACL 2009, pp. 683–691 (2009)
17. Lin, Z., Kan, M.Y., Ng, H.T.: Recognizing implicit discourse relations in the Penn Discourse Treebank. In: Proc. of EMNLP 2009, pp. 343–351 (2009)
18. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of COLT 1998, pp. 92–100 (1998)
19. Caruana, R.: Multitask learning: A knowledge-based source of inductive bias. In: Proc. of ICML 1993, pp. 41–48 (1993)
20. Plackett, R.L.: Karl Pearson and the chi-squared test. International Statistical Review / Revue Internationale de Statistique 51, 59–72 (1983)
21. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proc. of EMNLP 2006, pp. 120–128 (2006)

22. Loper, E., Bird, S.: NLTK: The natural language toolkit. In: Proc. of ACL 2002 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, pp. 63–70 (2002)
23. Magerman, D.M.: Statistical decision-tree models for parsing. In: Proc. of ACL 1995, pp. 276–283 (1995)
24. Klein, D., Manning, C.D.: Fast exact inference with a factored model for natural language parsing. In: Advances in Neural Information Processing Systems, vol. 15. MIT Press, Cambridge (2003)
25. Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., Webber, B.: The Penn Discourse Treebank 2.0 annotation manual. Technical report, University of Pennsylvania Institute for Research in Cognitive Science (2008)
26. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotatedcorpus of English: The Penn Treebank. Computational Linguistics 19, 313–330 (1993)
27. Okazaki, N.: Classias: A collection of machine-learning algorithms for classification (2009), http://www.chokkan.org/software/classias/
28. Zhou, Z.M., Xu, Y., Niu, Z.Y., Lan, M., Su, J., Tan, C.L.: Predicting discourse connectives for implicit discourse relation recognition. In: Proc. of COLING 2010 (Posters), pp. 1507–1514 (2010)
29. Louis, A., Joshi, A., Prasad, R., Nenkova, A.: Using entity features to classify implicit discourse relations. In: Proc. of the SIGDIAL 2010, pp. 59–62 (2010)