Caroline Sporleder

Manually vs. Automatically Labelled Data in Discourse Relation Classification: Effects of Example and Feature Selection

We explore the task of predicting which discourse relation holds between two text spans in which the relation is not signalled by an unambiguous discourse marker. It has been proposed that automatically labelled data, which can be derived from examples in which a discourse relation is unambiguously signalled, could be used to train a machine learner to perform this task reasonably well. However, more recent results suggest that there are problems with this approach, probably due to the fact that the automatically labelled data has particular properties which are not shared by the data to which the classifier is then applied. We investigate how big this problem really is and whether the unrepresentativeness of the automatically labelled data can be overcome by performing automatic example and feature selection.

1 Introduction

Machine learning approaches have been successfully applied to many areas of natural language processing (NLP). Usually the best results are achieved by *supervised* methods, in which a learner is trained on a set of manually labelled examples. However, manual annotation of training data is time-consuming and costly. In recent years, there has consequently been a shift towards techniques that reduce the annotation effort, either by careful selection of the examples to be annotated (*active learning* (Cohn et al., 1994)) or by mixing labelled and unlabelled data (e.g., via *co-training* (Blum and Mitchell, 1998)). In other cases, heuristics have been used to label some data automatically. One application for which this strategy has been suggested is the identification of discourse relations, such as CONTRAST or EXPLANATION, holding between text spans (Marcu and Echihabi, 2002).

Such discourse relations can be explicitly signalled by discourse connectives. For instance, in example (1) the CONTRAST relation between the two text spans (indicated by square brackets) is signalled by *but*. However, many discourse connectives are ambiguous between several relations, such as *since*, which can signal EXPLANATION (2a) but also a temporal relation (2b). Finally, many relations are not signalled by an explicit discourse marker at all, such as the RESULT relation in (3). Throughout this paper, we will call examples, in which the discourse relation is unambiguously signalled *marked*, and all other examples (i.e., those with an ambiguous marker or no marker at all) will be referred to as *unmarked*. Identifying the correct discourse relation in examples in which the relation is indicated by an unambiguous marker is fairly trivial; one just needs a list of such markers and the relations they map to. If the relation is signalled

by an ambiguous marker, the task becomes more difficult and involves disambiguating which of the relations that the marker can signal holds in a given example. But, because the number of distinct relations that an ambiguous marker can signal is typically very limited, the problem should still be relatively easy to solve. If no explicit discourse connective is present, however, the task becomes relatively challenging. In the absence of an explicit marker, a classifier has to rely on other cues, such as the lexical semantics of the words in the spans.

(1) [We can't win,] [**but** we must keep trying.]

(2) a. [I don't believe he's here] [**since** his car isn't parked outside.]

b. [She has worked in retail] [**since** she moved to Britain.]

(3) [The train hit a car on a level-crossing,] [it derailed.]

Marcu and Echihabi (2002) proposed that the problem could be addressed by utilising the existence of unambiguously marked relations in some examples to extract and automatically label training data for a classifier from a large unannotated corpus. The class label of an extracted example is then assigned on the basis of the unambiguous marker. Example (1), for instance, would be assigned the label CONTRAST. The marker is then removed and a classifier is trained on the automatically labelled data to distinguish between different relations even if no marker is present.

While this approach is very elegant and appealing, more recent studies (Murray et al., 2006; Sporleder and Lascarides, 2007) found evidence that training on automatically labelled data is of limited use for the identification of discourse relations in *unmarked* examples. These results seem to be independent of the classifier used, and it has been suggested (Sporleder and Lascarides, 2007) that the problem stems from the automatically labelled training data themselves, i.e., the fact that the automatically labelled data were originally unambiguously marked means that they are often different from unmarked data which makes it difficult for a classifier trained on the former to generalise to the latter.

In this paper, we aim to explore how big this problem really is. In particular, we explore whether a small set of (unmarked) manually annotated seed data can be exploited to overcome some of the problems associated with automatically labelled data. Such seed data can be used in at least two ways: (i) to select those automatically labelled examples which are most similar to unmarked data (i.e., to the seed data) and therefore, so we hypothesise, make good training examples for the classifier (*example selection*), and (ii) to determine which features generalise best from automatically labelled data to unmarked examples (*feature selection*). We look at both approaches and report the results in sections 4 and 5, respectively. First, however, we give a more detailed overview of the task and previous research (Section 2) and discuss the data and machine learning algorithms that were used in the experiments (Section 3).

2 Discourse Parsing and the Classification of Discourse Relations

Texts are not just random collections of sentences, they have internal structure, which is commonly referred to as *discourse structure*. There are numerous theories of discourse structure, e.g., *Rhetorical Structure Theory* (RST) (Mann and Thompson, 1987), *Discourse Representation Theory* (DRT) (Kamp and Reyle, 1993), *Segmented Discourse Representation Theory* (SDRT) (Asher and Lascarides, 2003) and *Discourse Lexicalised Tree Adjoining Grammar* (DLTAG) (Webber et al., 2003).

Typically, discourse is viewed as a hierarchical structure in which smaller discourse units (also known as *spans*) are connected by *discourse relations* (also known as *rhetorical relations*), such as EXPLANATION, RESULT, or CONTRAST, to form larger units which can then in turn be linked to other discourse units. For example, in (4), the second sentence relates to the first via a RESULT relation and the resulting larger unit links to the third sentence via a CONTINUATION relation (see Figure 1).

(4) a. The high-speed Great Western train hit a car on an unmanned level crossing yesterday.

b. It derailed.

c. Transport Police are investigating the incident.

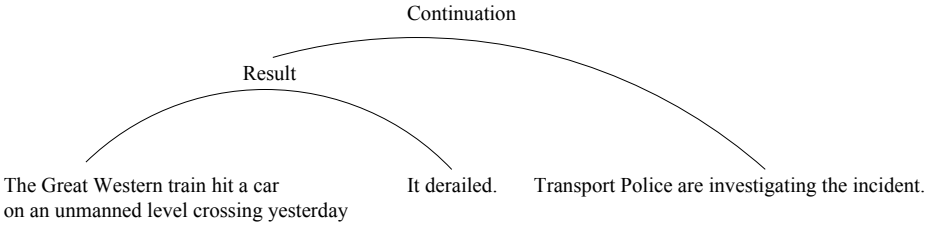

Figure 1: Discourse Structure of Example (4)

Knowledge of the discourse structure of a text would be beneficial for many applications, among them question-answering, information extraction, and text summarisation. As a consequence, there has been a lot of research on *discourse parsing*, i.e., determining the discourse structure of a text by automatic means. Most of the earlier work was rule-based, making use of hand-crafted rules that involved relatively deep semantic analyses (Hobbs et al., 1993; Asher and Lascarides, 2003). A second strand of work comprised rule-based systems that relied more heavily on surface cues than deep semantics (Corston-Oliver, 1998; Polanyi et al., 2004a,b; Le Thanh et al., 2004). With the advent of the first corpora that had been manually annotated with discourse structure, the focus shifted towards systems which employed machine learning techniques

to train a discourse parser on these resources (Marcu, 1999; Soricut and Marcu, 2003; Baldridge and Lascarides, 2005).

In this paper, we look at a sub-problem of full-blown discourse parsing, namely identifying the correct discourse relation between two adjacent sentences (*inter-sentential*) or between two clauses within a sentence (*intra-sentential*). We disregard the problem of determining relations in multi-sentence units[1] as well as the problem of determining the correct span boundaries and attachment sites. While these are interesting problems in themselves and, in practice, need to be solved together with the relation classification, the identification of discourse relations in unmarked examples is probably the most challenging sub-task of full discourse parsing. And this sub-task is particularly challenging when it involves the lower levels of a discourse tree, i.e., for relations between sentences or clauses.

Because the identification of discourse relations in unmarked examples is such a complex problem, requiring knowledge of lexical semantics and ideally also some form of world knowledge, machine learning approaches seem to be best suited for solving it. Trained systems achieve reasonably good performance (up to 60% F-score) (Marcu, 1999; Soricut and Marcu, 2003; Baldridge and Lascarides, 2005), but the necessity for annotated corpora is a big limitation. Such corpora are expensive to create and are therefore only available for very few languages. Consequently, there has also been research into how manual annotation of corpora can be avoided or reduced. Nomoto and Matsumoto (1999), for instance, propose an active learning solution for the task of identifying discourse relations between sentences.

Going one step further, Marcu and Echihabi (2002) present an approach which does not require any manual annotation effort at all. Instead they devise a scheme for labelling training data automatically by exploiting the fact that discourse relations are sometimes unambiguously marked. Such examples can be extracted from large unannotated corpora and automatically labelled with the appropriate relation. The discourse markers are then removed and a classifier is trained to identify the correct relation even *in the absence* of an unambiguous marker. This is necessary to adequately process the many examples that either contain no overt discourse marker or that contain a marker that is ambiguous between several relations.

Marcu and Echihabi (2002) applied their method to four relations from Mann and Thompson (1987), namely CONTRAST, CAUSE-EXPLANATION-EVIDENCE, CONDITION and ELABORATION (where CONTRAST and ELABORATION are supertypes for more specific relations in RST). Two types of non-relations (NO-RELATION-SAME-TEXT and NO-RELATION-DIFFERENT-TEXTS) were also included. Marcu and Echihabi identified a number of unambiguous discourse markers for these relations[2] and extracted examples of inter- and

[1] Discourse relations holding between multi-sentence units seem to have a somewhat different distribution than those holding between individual sentences. For example, relations such as TOPIC-SHIFT and SUMMARY are more frequent between multi-sentence segments whereas relations such as EXPLANATION or CONTRAST tend to hold between relatively short spans. Consequently, it has been suggested to implement two different processing strategies for higher and lower level discourse structure (Marcu, 1997).

[2] The non-relations are extracted by randomly choosing pairs of non-adjacent sentences. Because these sentences are non-adjacent it is assumed that no discourse relation holds between them.

intra-sentence relations[3] from a 40 million sentence corpus. They obtained between 900,000 and 4 million examples per relation. The discourse markers were then removed from the extracted data and a Naive Bayes classifier was trained to distinguish between different relations on the basis of co-occurrences between pairs of words, with one word in the pair coming from the left span and the other from the right. Marcu and Echihabi (2002) mainly tested their method on a set of automatically labelled data, i.e., examples which originally contained an unambiguous discourse marker which was used for labelling the example with the gold standard relation and then removed before testing. For this data set they report an accuracy of 49.7% for the six-way classifier. They also tested several binary classifiers (distinguishing between ELABORATION and each of the other relations) on a set of *unmarked* examples. However, they do not report the accuracy or F-score for these experiments, only the recall on the non-ELABORATION relation, which lies between and 44.74% and 69.49%

In later work, Sporleder and Lascarides (2007) investigated more extensively how useful automatically labelled examples are in practice for classifying discourse relations in unmarked examples. They chose five relations from SDRT's inventory of relations (Asher and Lascarides, 2003): CONTRAST, RESULT, EXPLANATION, SUMMARY and CONTINUATION. These relations were selected because for each of them there are discourse markers which signal them unambiguously but they also frequently occur *without* a discourse marker, making it beneficial to be able to determine them automatically if no marker is present. Three corpora were used to extract training data: the British National Corpus (BNC, 100 million words), and two corpora from the news domain — the North American News Text Corpus (350 million words) and the English Gigaword Corpus (1.7 billion words). The number of extracted examples ranged from 8,500 for CONTINUATION to just under 7 million for CONTRAST. In addition, an annotated set of unmarked examples was created by manually labelling data from the RST Discourse Treebank (RST-DT) (Carlson et al., 2002) with SDRT relations. This data set contained 1,051 examples with roughly equal numbers of examples for each of the relations (with the exception of SUMMARY which occurs relatively infrequently in the RST-DT, so only 44 examples were found).

Sporleder and Lascarides (2007) carried out three experiments: (1) training and testing on automatically labelled data (i.e., data in which the relation was originally marked by a discourse marker which was then removed), (2) training on automatically labelled data and testing on unmarked, manually labelled data, and (3) training and testing on manually labelled unmarked data. To investigate whether there are any classifier-specific differences in performance, they employed two different classifiers. The first was a re-implementation of Marcu and Echihabi's (2002) Naive Bayes model and used only lexical features. The other employed a variety of linguistically motivated features (see Sporleder and Lascarides (2007, 2005)) and used the BoosTexter machine learning framework (Schapire and Singer, 2000).

[3] Where the inter-sentence relations involve adjacent sentences. The boundaries of the spans participating in the relation are determined using a set of heuristics based on surface cues.

It was found that training and testing on automatically labelled data led to reasonable results (61% accuracy for the BoosTexter model, 42% for the Naive Bayes model). This suggests that discourse relations are, in principle, learnable from automatically labelled data. However, when the classifiers trained in this way were applied to unmarked examples the performance of both of them dropped to around 26% accuracy. This was just above the baseline of choosing a relation randomly (20% accuracy), though the difference was statistically significant. By comparison, training the BoosTexter model on just 500 manually labelled, unmarked examples led to a noticeably higher accuracy of 40%. A learning curve experiment revealed that just under 140 manually labelled training data were enough to rival the performance that was obtained by training on 72,000 automatically labelled data. Similar findings are reported by Murray et al. (2006) who investigated the learnability of discourse relations in speech.

These results suggest that, (i) it is *possible* to learn discourse relations from automatically labelled data (because training and testing on automatically labelled data led to reasonable results), but (ii) classifiers trained in this way do not generalise well on unmarked data. The fact that both classifiers dropped to similar performance levels indicates furthermore that this is not predominantly a problem with the feature space or the learning framework but that the problem probably stems from the training data itself: Automatically labelled examples, in which the relation was originally marked, may simply be too different from unmarked examples to make good training material for the latter.[4] For instance, a typical marked example of a given relation may exhibit structural properties that are very different from the properties of a typical unmarked example for the same relation. Sporleder and Lascarides (2007) carried out a preliminary study of various linguistic properties of marked and unmarked examples and found some notable disparities, such as a significant variation in span length between marked and unmarked instances of the RESULT relation, and differences in the distribution of part-of-speech tags for the CONTRAST, EXPLANATION, and RESULT relations. In some cases these differences meant that features which were highly predictive of a relation in the marked examples were not predictive of the same relation in the unmarked examples. For instance, CONTINUATION was nearly always holding inter-sententially in the marked examples but occurred more frequently as an intra-sentential relation in the unmarked examples. The predominantly inter-sentential distribution of CONTINUATION in the automatically labelled training data caused the BoosTexter model to learn a decision rule which only predicted CONTINUATION if a relation was holding inter-sententially. While this rule had a relatively high accuracy when applied to the (originally marked) automatically labelled data, it led to a fairly low accuracy on the unmarked data.

[4] An alternative explanation would be that the automatically labelled examples are just too noisy, where the noise comprises mislabelled relations as well as misplaced span boundaries. However, Sporleder and Lascarides (2007) found that their extraction method was fairly accurate, with just 2% of the examples in a manually checked sample containing an error.

3 Data and Machine Learners

In this paper, we investigate whether a small set of manually labelled data can be exploited to automatically select good training data and suitable features which will help to overcome some of the problems associated with training on automatically labelled data. In all experiments we use the same data and the same classifiers that were used by Sporleder and Lascarides (2007).

As mentioned in the previous section, the automatically labelled data was extracted from three corpora: the BNC, the American News Text Corpus and the English Gigaword Corpus. The extracted data covers five relations from SDRT (Asher and Lascarides, 2003): CONTRAST, EXPLANATION, RESULT, SUMMARY, and CONTINUATION. Because the data was highly skewed (with 7 million extracted for CONTRAST but only 8,500 for CONTINUATION) which causes problems for most machine learners, a smaller data set of approximated 72,000 examples was created with roughly uniform distributions across the five relations.

The manually labelled data set contains 1,051 unmarked examples. These were taken from the RST-DT and manually mapped to the corresponding SDRT relations. The distribution of relations in this data set was also approximately uniform, except for SUMMARY of which only 44 unmarked examples were found in the RST-DT.

We used two different classifiers in the experiments. We deliberately chose two classifiers which are fairly different, both with respect to the machine learning technique they use and with respect to their feature space. The first was a re-implementation of the Naive Bayes model proposed by Marcu and Echihabi (2002). This model assumes that the relation that holds between two spans can be determined on the basis of co-occurrences between words.

Let r_i be the discourse relation that holds between two spans W_1 and W_2. The model assumes that r_i can be determined on the basis of the word pairs in the Cartesian product over the words in the two spans: $(w_i, w_j) \in W_1 \times W_2$. The model is derived as follows: Given the assumption in the word pair model, the most likely relation is given by $\underset{r_i}{\text{argmax}}\ P(r_i|W_1 \times W_2)$. According to Bayes rule:

$$P(r_i|W_1 \times W_2) = \frac{P(W_1 \times W_2|r_i)P(r_i)}{P(W_1 \times W_2)} \tag{5}$$

Since for any given example $P(W_1 \times W_2)$ is fixed, the following holds:

$$\underset{r_i}{\text{argmax}}\ P(r_i|W1 \times W_2) = \underset{r_i}{\text{argmax}}\ P(W_1 \times W_2|r_i)P(r_i) \tag{6}$$

We estimate $P(r_i)$ via maximum likelihood on the training set. And to estimate $P(W_1 \times W_1|r_i)$ we assume that all word pairs in the Cartesian product are independent, i.e.:

$$P(W_1 \times W_2|r_i) \approx \prod_{(w_i,w_j)\in W_1 \times W_2} P((w_i, w_j)|r_i) \tag{7}$$

To estimate the probability of a word pair (w_i, w_j) given a relation r_i, we use maximum likelihood estimation and Laplace smoothing. We converted all words in the spans to lower case but — to stay faithful to Marcu and Echihabi (2002) — we did not apply any other pre-processing, such as stemming.

The second model is more complex. It uses a variety of shallow linguistic features (Sporleder and Lascarides, 2005). To combine these features into a classifier we used BoosTexter (Schapire and Singer, 2000), which integrates a boosting algorithm with simple decision rules and allows a variety of feature types, such as nominal, numerical or text-valued features. Text-valued features can, for instance, encode sequences of words or part-of-speech tags. BoosTexter applies n-gram models when forming classification hypotheses for these features (i.e., it tries to detect n-grams in the sequence which are good predictors for a given class label).

We implemented 41 linguistically motivated features, roughly falling into six classes:

- **positional features:** whether the relation holds inter- or intra-sententially, and the position of the example relative to the preceding and following paragraph boundaries
- **length features:** span length
- **lexical features:** words, lemmas, stems, and their overlap, WordNet (Fellbaum, 1998) classes
- **part-of-speech features:** part-of-speech tags
- **temporal features:** finiteness, modality, aspect, voice and negation of the verb phrases
- **cohesion features:** pronoun distribution, presence or absence of ellipses

As some of these features rely on stems, lemmas, and syntactic chunking, we preprocessed the examples with the Porter stemmer (Porter, 1980), the RASP toolkit[5] (Minnen et al., 2001) and the Charniak parser (Charniak, 2000). For a more detailed description of the features and their motivation see Sporleder and Lascarides (2007).

4 Automatic Example Selection

The results reported by Sporleder and Lascarides (2007) suggest that training on automatically labelled data alone does not lead to a satisfactory performance on unmarked examples. This may be because automatically labelled data are derived from examples in which the discourse relation was originally unambiguously marked and these marked examples may be structurally too different from unmarked examples to make good training material for a classifier that is then applied to unmarked data. However, it may be that not all automatically labelled instances are equally bad training material. Marked and unmarked examples are not per se structurally different. Sometimes

[5]Downloadable from `http://www.informatics.susx.ac.uk/research/nlp/rasp/` (20.4.2007).

a discourse marker can be added or removed from a pair of spans without rendering the example infelicitous, as in (8) below. An automatically labelled example (8a) would be indistinguishable from a manually labelled example (8b) and should thus make an equally good training instance. Moreover, even in cases where there is no complete one-to-one correspondence between marked and unmarked examples, the resulting automatically labelled examples are not necessarily useless; as long as they are not too different from unmarked examples a classifier might still be able to learn something from them.

(8) a. She doesn't make bookings **but** she fills notebooks with itinerary recommendations.
b. She doesn't make bookings, she fills notebooks with itinerary recommendations.

Hence, the question is whether it is possible to automatically select those examples that are useful and informative for the classifier. In most automatic example selection approaches, it is assumed that informative examples are those about whose class label the learner is least certain (*uncertainty sampling* (Lewis and Catlett, 1994)). This is the idea behind active learning, where the aim is to reduce the annotation effort by selecting those examples for annotation whose class label cannot yet be predicted confidently, which often means that a smaller training set obtained via active learning leads to models with comparable performance to those that are trained on much larger randomly selected training sets (e.g., Baldridge and Osborne (2004)). In our case, the aim is somewhat different: instead of selecting those examples about which the learner is least certain, we need to select those examples which are most useful for training a classifier that can predict the discourse relations in *unmarked examples*.

One way of doing this would be by using a small manually labelled set of unmarked examples to evaluate the effect of adding and deleting a particular example (or a subset of examples) from the marked, automatically labelled training set. This way the set of automatically labelled examples could be searched for a good subset for training, i.e., a subset which is 'representative' of the unmarked examples that we wish to model. This is called a *wrapper approach* (Blum and Langley, 1997). Wrapper approaches use the induction algorithm itself (in our case the BoosTexter or Naive Bayes model) to determine which examples to include in the final training set. An alternative to the wrapper approach is a *filter approach*, where the training examples are selected independently of the induction algorithm, using some other measure to decide which examples to include.

It has been argued that wrapper methods often lead to better results because they take the bias of the main induction algorithm into account (see Kohavi and John (1997) in the context of feature selection). However, they have the disadvantage of being computationally more expensive than filter methods. Therefore, we opted for a filter approach. We made the simplifying hypothesis that "good" training examples are those which are most similar to manually labelled examples. We could apply a pre-defined similarity criterion, e.g., based on feature vector overlap, to identify those automatically labelled examples which are most similar to the examples in our manually labelled development set. However, we decided to take a different approach and train a classifier

to distinguish between automatically and manually labelled examples. We then applied this classifier to the automatically labelled data and selected those examples which the classifier labels as "manual" with the highest probability.

Note that we require a manually labelled data set for our example selection. This means that—from a practical perspective—it is not enough that we can show that training on automatically selected examples leads to a better performance on the manually labelled test set than training on randomly selected examples. Instead, to justify the use of automatically labelled examples, we have to show that using the selected examples leads to a better performance than training on the manually labelled development set alone. That is, we should show that it is possible to bootstrap a classifier by starting with a small set of manually annotated examples and then use these to select further examples from a pool of automatically labelled data and thereby enhance the performance of the original classifier. The baseline for the experiments in this section is therefore the performance of our two models (i.e. the BoosTexter and the Naive Bayes model) when trained on the development set alone.

For comparison, we also investigate what happens if a manually labelled data set is mixed with *randomly* selected automatically labelled data. These experiments are reported in the following section. Section 4.2 then discusses the 'similarity-based' example selection method in more detail, and examines its effects on the performance of the models.

4.1 Random Example Selection

To determine whether the performance of training on a small set of manually labelled examples can be improved upon by adding randomly selected automatically labelled examples, we split the manually labelled set in two halves of 525 instances, one for testing and the other to merge with the automatically labelled examples for training. We then created 15 sets of randomly selected automatically labelled examples of increasing sizes, ranging from 20% of the manually labelled training set (105 automatically labelled instances) to 300% (1,575 instances). The distribution of relations was kept uniform in each sample. We merged each sample with our manually labelled training set, trained the classifiers and then tested on the unseen test set. Then we swapped the manually labelled test and training sets, re-trained and re-tested and averaged the results. Figure 2 shows the learning curves obtained in this way.

For the Naive Bayes word pair model, it can be observed that adding automatically labelled examples generally improves the accuracy. This is due to the fact that this model relies on word features alone and is thus particularly sensitive to sparse data problems. However, big variations in accuracy begin to occur at sampling rates above 100%, where the automatically labelled examples start to dominate the training set. That is, the Naive Bayes model seems to be fairly sensitive to the quality of the training data. Despite this variation, it can be observed that the curve flattens at around 25% accuracy. This level is first achieved for a 160% sampling rate (1,365 training instances overall), and adding further automatically labelled examples does not significantly im-

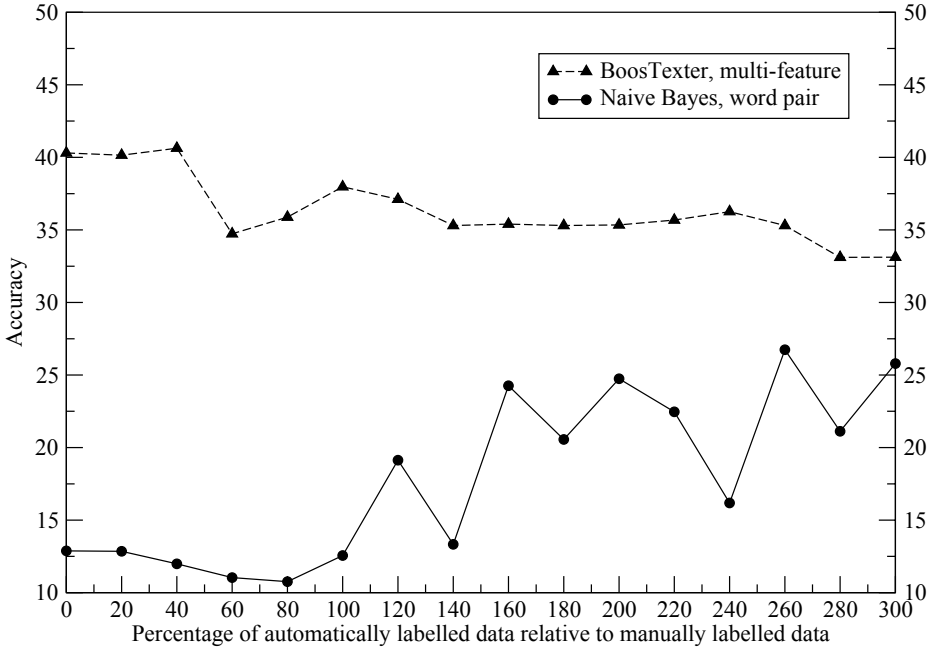

Figure 2: Learning curve for mixing manually and automatically labelled data, averaged over two manually labelled training and test sets

prove the result. Note that training the model on the whole set of automatically labelled examples (around 72,000 instances) and then testing on the manually labelled data also led to an accuracy of around 25%. So it looks like this is the maximum that can be obtained with this model when training on mostly automatically labelled examples and testing on unmarked data.

For the BoosTexter model the situation is different. This model performs relatively well when trained on a small set of manually labelled data and adding randomly selected automatically labelled examples generally decreases the accuracy. At a sampling rate of 300% the accuracy is still around 33% (compared with 40.3% accuracy when trained only on manually labelled data) but, if more automatically labelled examples were added, one would expect this to fall further to around 25%, as this is the level of performance achieved by training on the whole set of automatically extracted examples. Note that there is less variance in the learning curve for the BoosTexter model.

This suggests that this model is better at making the most of each training set.

4.2 Using Machine Learning to Filter Examples

In the previous section, we saw that augmenting an unmarked, manually labelled seed training set with randomly selected, automatically labelled examples hurts performance, at least for the better performing BoosTexter model, which is less sensitive to sparse data. The fact that automatically labelled training examples can hurt performance on the manually labelled test set provides further evidence that the two data sets are linguistically quite different. Given the potential syntactic and semantic differences between them, it makes sense to use machine learning to estimate which of the automatically labelled examples are similar to the manually labelled set, and to use these to augment the seed training set.

To this end, this section presents a more sophisticated example selection method than choosing them randomly. We only ran this experiment for the BoosTexter model as the Naive Bayes word pair model never reached an accuracy of more than 26% in the previous experiments, hence it is unlikely that a more sophisticated sampling method will cause it to outperform the 40% accuracy obtained by the BoosTexter model when trained on the manually labelled data. Moreover, boosting the training set with randomly selected automatically labelled examples actually hurt the performance of the BoosTexter model, and we want to see if other methods of selection can reverse this.

To determine which instances are similar to unmarked examples, we used BoosTexter to train a classifier to distinguish between manually and automatically labelled examples. The training set for this classifier was created by merging half of the manually labelled examples with an equal number of randomly selected automatically labelled examples (with a uniform distribution of rhetorical relations) and replacing the original class labels by a new label encoding whether an instance came from the manually or the automatically labelled set. In a similar way we also created a test set (using the other half of the manually labelled data) so that we could determine how well the classifier could distinguish between the two types of data. We kept the original feature space for our new binary, manual-vs.-automatic classifier.

Table 1 shows the results of applying the binary classifier to the test set. In this table we also report the number of instances for which a given class was predicted (*pred.*), the number of instances labelled with a given class in the gold standard (*GS*), and the number of instances which were correctly labelled by the classifier (*correct*). It can be observed that the accuracy and F-score of the classifier are relatively high at 74.95% and 73.47%, respectively. This is well above the 50% accuracy baseline of choosing a class randomly. Thus it seems that automatically and manually labelled examples can indeed be distinguished to some extent in our feature space. Note, however, that the classifier predicts "manual" more frequently (774 times) than "automatic" (276 times), hence some of the automatically labelled examples seem to be similar enough to manually labelled examples to be assigned to the "manual" class. The question is whether this

set of examples can be used to boost the performance of the discourse relation classifier when added to a small set of manually labelled examples.

Class	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score	pred.	GS	correct
manual	n/a	66.93	98.67	79.76	774	525	518
automatic	n/a	97.46	51.24	67.17	276	525	269
all	74.95	82.20	74.96	73.47	n/a	1,050	787

Table 1: Testing the automatic-vs.-manual classifier

To test this we applied the automatic-vs.-manual classifier to the set of automatically extracted training examples. Around 66% of the instances were classified as "manual" by our binary classifier. The confidence of the classifier in the class label is reflected in the weight it assigns to it. When deciding which automatically extracted examples to add to the manually labelled training set, we could choose those which are most confidently predicted to belong to the "manual" class, i.e., those with highest weight for that class. However, we took a slightly different approach: instead of choosing the absolutely highest scoring examples, we selected randomly from the top 10%. The motivation for this is that, while we want to select examples which are similar to manually labelled instances, we also want to select examples which are informative for the learner, i.e., those from which something new can be learnt. Randomly sampling from the top 10% of examples ensures that we select examples which are similar to the manually labelled instances but we do not necessarily select the *most* similar ones.

Using this strategy we selected 525 automatically labelled examples, making sure that the distribution of rhetorical relations was uniform. We merged this set with the 525 manually labelled examples that we had trained our binary, automatic-vs.-manual classifier on. The resulting set was then used as training material for the relation classifier and the trained model was tested on the other half of the manually labelled data (which was not used in the example selection process).

Table 2, which is taken from Sporleder and Lascarides (2007), shows the result of training on 535 manually labelled examples alone (averaged over two runs). This is the baseline that we would hope to beat by adding carefully selected automatically labelled training data to the manually labelled seed data set. Table 3 shows the results of training on 525 manually labelled data and an equal amount of automatically selected automatically labelled data. It can be observed that our sampling strategy does not lead to an improved performance over training on the manually labelled data alone; on the contrary the accuracy drops by around 5%. For comparison, Table 4 shows the results of randomly selecting 525 automatically labelled examples (averaged over five random samples). Random sampling actually leads to a slightly better performance than our machine learning based sampling strategy, though this difference is not significant ($\chi^2 = 1.54, DoF = 1, p <= 0.22$). There could be several reasons for this. First, it is

Relation	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score
CONTINUATION	n/a	36.78	36.85	36.77
RESULT	n/a	38.53	46.32	41.99
SUMMARY	n/a	13.75	3.64	5.63
EXPLANATION	n/a	49.80	50.15	49.85
CONTRAST	n/a	36.70	32.21	34.19
all	40.30	35.11	33.83	33.69

Table 2: Baseline: Training and Testing on Manually Labelled Data, 5 times 2-fold cross-validation averaged

possible that our underlying assumption that automatically labelled examples which are similar to manually labelled ones make good training material is wrong. Second, it could be that this hypothesis is valid but that the problem lies with our automatic-vs.-manual classifier, i.e., it may be that this classifier is simply not accurate enough (though we have found it to achieve an accuracy of above 75%).

Relation	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score
CONTINUATION	n/a	38.06	39.23	38.64
RESULT	n/a	29.81	23.31	26.16
SUMMARY	n/a	7.14	4.55	5.56
EXPLANATION	n/a	40.72	50.75	45.19
CONTRAST	n/a	32.50	31.94	32.22
all	35.24	29.64	29.95	29.55

Table 3: Training on 50% manually labelled and 50% automatically selected automatically labelled examples

5 Using Manually Labelled Data for Automatic Feature Selection

In the previous section, we experimented with automatic example selection and found that adding training examples selected in this way to a manually annotated seed set does not lead to any improvements compared to training on the manually labelled data alone. On the contrary, adding automatically labelled examples decreases the accuracy of the classifier and it does not seem to matter whether the examples are selected randomly or using the similarity-based approach. In this section, we explore the effect of automatic feature selection. Sporleder and Lascarides (2007) argued that one reason

Relation	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score
CONTINUATION	n/a	38.27	38.93	38.54
RESULT	n/a	37.06	37.07	37.06
SUMMARY	n/a	12.67	18.18	14.84
EXPLANATION	n/a	43.62	46.27	44.77
CONTRAST	n/a	39.57	31.55	34.64
all	37.97	34.24	34.40	33.94

Table 4: Training on 50% manually labelled and 50% randomly selected automatically labelled examples, averaged over 5 sampling runs

why a classifier that was trained on automatically labelled data did not generalise well to unmarked examples, could be that —due to structural differences between the two types of data— some features that are predictive for a given relation on one type of data are not predictive on the other type of data and vice versa. One way to overcome this problem could be by automatically selecting those features which maximise the classifier's performance on unmarked data. This can be done by testing individual features and feature combinations on a small manually annotated seed data set (the *development* set) and selecting only the best performing ones.

The feature selection was only performed for the BoosTexter model, since that was found to consistently lead to better results than the Naive Bayes model. We employed a greedy, wrapper-based feature selection strategy. The manually labelled data was split into two parts with equal proportions of the five relations: 20% (i.e., 208 examples) were used as a development set in the feature selection process, the remaining 80% (843 examples) were used for testing. The selection process was started off by training a one-feature classifier for each of the 41 features in our complete feature set. The classifiers were trained on the complete set of automatically labelled examples and then tested on the 208 manually labelled examples in the development set. The best performing feature was then selected. In the next round each of the remaining 40 features was added individually to the best performing feature from the previous round. The resulting 40 2-feature classifiers were trained and tested again, and the best performing 2-feature set was used as the basis to which new features were added in the next round and so on. The feature selection process stopped when adding features did not lead to any improvements in accuracy on the development set. Once the features had been selected, a classifier was trained on the automatically labelled data using only the selected features. The classifier was then tested on the remaining 80% of the manually labelled test set. To abstract away from possible idiosyncracies of the test and development set, we ran the experiment five times, each time with a different 80:20 split of test and development data. The results of the five experiments (*Run 1* to *Run 5*) are

reported in Table 5. The table lists the number of features selected (# *Features*), the accuracy that was achieved with the selected features on the development set (*Devel. Acc.*) and on the unseen test set (*Test Acc.*), and the test set accuracy that was obtained with the whole feature set (*Test Acc. all feat.*). From the results it is evident that feature selection does not have a noticeable positive effect on the performance of the classifier. While the test set accuracy with the reduced feature set is often somewhat higher than with the full set (e.g., for Run 1), this difference was not found to be significant in any of the cases. In those cases where training on the full feature set leads to better results, the difference was also not significant.

	# Features	Devel. Acc.	Test Acc.	Test Acc. all feat.
Run 1	3	30.29%	24.08%	24.04%
Run 2	5	35.10%	24.32%	25.27%
Run 3	3	29.33%	25.86%	25.86%
Run 4	4	33.17%	27.16%	25.86%
Run 5	3	31.51%	26.68%	25.84%
Avg.	3.6	31.88%	25.62%	25.37%

Table 5: Feature Selection on a Manually Labelled Development Set

Table 5 also shows that the feature selection algorithm overfits on the development set, i.e., the performance on the development set is noticeably higher than on the test set. This might be due to the somewhat simplistic selection algorithm, which only stops adding features when *no* accuracy gain can be obtained anymore. Having a more sophisticated stopping criterion would probably reduce this overfitting effect. However, the number of features selected by our algorithm is generally very small (3.6 on average), so stopping earlier would result in only one or two selected features. This is unlikely to lead to any significant improvements as the selected features vary a lot between different runs, (i.e., they depend very much on the data sets that are used), which becomes evident from Table 6 in which the features that were selected in the five runs are listed.

Run 1	dist. prev. paragraph, tense info left, adjective lemma overlap
Run 2	content word lemmas left, verb lemma overlap, dist. prev. paragraph, noun lemmas left, pronouns 2P right
Run 3	words left, span length right, ellipsis left
Run 4	words left, word overlap, noun WordNet classes overlap, noun lemma overlap
Run 5	content word lemma overlap, pronouns 3P left, pronouns 2P right

Table 6: Features Selected on Different Runs (in order of selection)

There is relatively little overlap between the different runs with respect to the selected features;[6] no feature was selected in every runs and only three features were selected in more than one run: the distance from the preceding paragraph boundary (*dist. prev. paragraph*, two runs), the number of second person pronouns in the right span (*pronouns 2P right*, two runs), and the words in the left span (*words left*, two runs). To some extend this variation can be explained by the fact that the lexical features are often not very different from each other (e.g., *words left* vs. *content word lemmas left*), so it may be a matter of coincidence which one is chosen first and once this feature has been chosen it makes no sense to add the other one anymore. However, there is also much variation between different *types* of features. For example, in Run 5, predominantly cohesion features are chosen (*pronouns 2P right*, *pronouns 3P left*) which do not occur much in the other runs. One pattern that does arise, however, is that lexical features generally seem to be quite important, hence all runs include at least one lexical feature. Also, information about the left span seems to be more important than information about the right span. This observation was also reported by Sporleder and Lascarides (2005) who explored which features performed best on an *automatically labelled* test set (hence this is not a difference between manually and automatically labelled data). From a human discourse processing perspective it seems plausible that the left span should contain more information about the upcoming discourse relation: it makes the information easier to process than if the signalling is delayed until the right span.

It is also interesting to note that the features which were identified as potentially problematic by Sporleder and Lascarides (2007) because they encode properties on which marked and unmarked examples tend to differ, tend *not* to be selected. Such features are, for example, those that encode span length, part-of-speech tags, or whether the relation holds inter- or intra-sententially.[7] We performed a number of control experiments in which we ran the feature selection on an automatically labelled development set, and found that those features do get selected in that case. In other words, it looks like the feature selection process is able to identify and avoid the most problematic features, i.e., those which do not generalise from marked to unmarked data. However, this does not seem to be enough to reliably and significantly boost performance.

6 Conclusion

Recent research by Sporleder and Lascarides (2007) and Murray et al. (2006) found evidence that Marcu and Echihabi's (2002) suggestion of using automatically labelled data to train a classifier to determine discourse relations in unmarked examples does not work very well in practice. The likely reason for this is that the two types of examples are too dissimilar, i.e., automatically labelled examples, in which the relation was orig-

[6] To check in how far this variation is due to the size of the development set, we ran a similar experiment with a 50:50 split of the manually labelled data into test and development set (i.e., with 525 examples in each of the sets). For these data set, the algorithm generally only selected on or two features and there was still a fair amount of variation.

[7] The only "problematic" feature that gets selected is *span length right* which was selected in Run 3.

inally marked by an unambiguous discourse connective, are simply not representative of the unmarked examples to which the classifier is applied. In this paper, we investigated how fundamental this problem really is and whether a small set of manually labelled seed data (of unmarked examples) might be harnessed to overcome the unrepresentativeness of automatically labelled examples. In particular, we looked at whether such seed data could be used (i) to select automatically labelled examples which are similar to the unmarked data we wish to model and hence make good training material for the classifier, and (ii) to select features which generalise well from the marked, automatically labelled data to the unmarked data.

We found problems with both approaches. For a classifier which is very sensitive to sparse data, like the word-based Naive Bayes model proposed by Marcu and Echihabi (2002), boosting a small manually labelled seed data set with automatically labelled examples helps to improve performance. But, for our re-implementation of this model, we found that the accuracy does not seem to improve much beyond 26% for a 5-way classification task, no matter how much automatically labelled training data is added. This is hardly an acceptable performance level for any real-world application.[8] For models which are less afflicted by sparse data problems, like our multi-feature BoosTexter model, adding automatically labelled data to a manually labelled seed set actually harms performance, and this effect is already noticeably for relatively small amounts of added data. Moreover, it does not seem to make much difference whether the automatically labelled examples are selected randomly or via a more sophisticated, similarity-based selection strategy.

The feature selection experiments lead to similarly sobering results. While training on a carefully selected reduced feature set often led to somewhat better results than training on the full set, this difference was never significant. Moreover, the feature selection was found to be fairly dependent on the data sets that were used. One positive aspect was that features modelling linguistic properties that Sporleder and Lascarides (2007) identified as varying a lot between marked and unmarked data (e.g., span length, inter- vs. intra-sentential relations, part-of-speech tags) tended not to be chosen. The selected features were mainly lexical, thus it seems that lexical features generalise best from the marked to the unmarked case.

Given these results, it is not clear that automatically labelled data can be turned into a valuable resource for this task. It may be possible that sophisticated lexical features can be developed that do in fact generalise from automatically labelled to unmarked data. But a better strategy would probably be to invest resources in the creation of manually annotated data, e.g., corpora annotated with discourse such as the RST-DT,[9] the *Penn Discourse Treebank*,[10] or the *Potsdam Commentary Corpus*,[11] and in the development of

[8] It should be noted that training on automatically labelled data does not always lead to unacceptable results. While machine learning systems that are trained on such data generally perform less well than those trained on manually labelled data, adding automatically labelled instances to a small manually labelled set can sometimes boost performance, as in co-training (Blum and Mitchell, 1998).

[9] `http://www.ldc.upenn.edu/Catalog/LDC2002T07.html`.

[10] `http://www.seas.upenn.edu/~pdtb/`.

[11] `http://www.ling.uni-potsdam.de/cl/cl/res/forsch_pcc.html`.

good classifiers which can make the most of even a small amount of training data.

Acknowledgements

Part of this work was carried out at the University of Edinburgh, funded by EPSRC grant number GR/R40036/01. I am grateful to Alex Lascarides for many interesting discussions on this topic.

References

Asher, N. and Lascarides, A. (2003). *Logics of Conversation*. Cambridge University Press.

Baldridge, J. and Lascarides, A. (2005). Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*.

Baldridge, J. and Osborne, M. (2004). Active learning and the total cost of annotation. In *Proceedings of Empirical Approaches to Natural Language Processing (EMNLP)*.

Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory (COLT)*.

Carlson, L., Marcu, D., and Okurowski, M. E. (2002). RST Discourse Treebank. Linguistic Data Consortium.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Assocation for Computational Linguistics*, pages 132–139, Seattle, WA.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

Corston-Oliver, S. H. (1998). Identifying the linguistic correlates of rhetorical relations. In *Proceedings of the ACL Workshop on Discourse Relations and Discourse Markers*, pages 8–14.

Fellbaum, C., editor (1998). *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Hobbs, J. R., Stickel, M., Appelt, D., and Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

Kamp, H. and Reyle, U. (1993). *From Discourse To Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence Journal*, 1–2:273–324.

Le Thanh, H., Abeysinghe, G., and Huyck, C. (2004). Generation discourse structures for written text. In *Proceedings of COLING-04*, pages 329–335.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 148–156.

Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, ISI, Los Angeles, CA.

Marcu, D. (1997). *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD thesis, Department of Computer Science, University of Toronto.

Marcu, D. (1999). A decision-based approach to rhetorical parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 365–372.

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of ACL-02*, pages 368–375.

Minnen, G., Carroll, J., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Murray, G., Taboada, M., and Renals, S. (2006). Prosodic correlates of rhetorical relations. In *Proceedings of the HLT-NAACL ACTS Workshop*.

Nomoto, T. and Matsumoto, Y. (1999). Learning discourse relations with active data selection. In *Proceedings of EMNLP-99*.

Polanyi, L., Culy, C., van den Berg, M., Thione, G. L., and Ahn, D. (2004a). A rule based approach to discourse parsing. In *Proceedings of the 5th SIGDIAL Workshop in Discourse and Dialogue*, pages 108–117.

Polanyi, L., Culy, C., van den Berg, M., Thione, G. L., and Ahn, D. (2004b). Sentential structure and discourse parsing. In *Proceedings of the ACL-04 Workshop on Discourse Annotation*.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14:130–137.

Schapire, R. E. and Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Sporleder, C. and Lascarides, A. (2005). Exploiting linguistic cues to classify rhetorical relations. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-05)*.

Sporleder, C. and Lascarides, A. (2007). Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*. to appear.

Webber, B. L., Knott, A., Stone, M., and Joshi, A. (2003). Anaphora and discourse structure. *Computational Linguistics*, 29(4):545–588.