Implicit Discourse Relation Recognition by Selecting Typical Training Examples

Xun Wang[1] *Sujian Li*[1*] *Jiwei Li*[1] *Wenjie Li*[2]
(1) Key Laboratory of Computational Linguistics, Peking University, Ministry of Education, CHINA
(2) Department of Computing, Hong Kong Polytechnic University
{xunwang,lisujian,bdlijiwei}@pku.edu.cn, cswjli@comp.polyu.edu.hk

ABSTRACT

Implicit discourse relation recognition is a challenging task in the natural language processing field, but important to many applications such as question answering, summarization and so on. Previous research used either artificially created implicit discourse relations with connectives removed from explicit relations or annotated implicit relations as training data to detect the possible implicit relations, and do not further discern which examples are fit to be training data. This paper is the first time to apply a different typical/atypical perspective to select the most suitable discourse relation examples as training data. To differentiate typical and atypical examples for each discourse relation, a novel single centroid clustering algorithm is proposed. With this typical/atypical distinction, we aim to recognize those easily identified discourse relations more precisely so as to promote the performance of the implicit relation recognition. The experimental results verify that the proposed new method outperforms the state-of-the-art methods.

KEYWORDS: Discourse relation recognition, single centroid clustering, implicit discourse relation.

* Corresponding author.

Proceedings of COLING 2012: Technical Papers, pages 2757–2772,
COLING 2012, Mumbai, December 2012.

1 Introduction

It is widely agreed that sentences/clauses are usually not understood in isolation, but in relation to their neighbouring sentences/clauses. The task of discourse relation recognition is to identify and label the relations between sentences/clauses, which is fundamental to many natural language processing applications such as question answering, automatic summarization and so on.

Discourse relations, such as comparison and causal relations, can be divided into explicit and implicit relations by the presence or absence of discourse connectives (e.g., *but*, *because* et. al.). Previous study indicates that the presence of discourse markers can greatly help relation recognition and the most general senses (i.e., comparison, contingency, temporal and expansion) can be disambiguated with 93% accuracy based solely on the discourse connectives (Pitler et al., 2008). On the other hand, the absence of explicit textual cues makes it very difficult to identify the implicit discourse relations. Thus, recently discourse relation recognition research puts more efforts to meet the challenges in implicit discourse relation recognition.

Existing work mainly focused on exploiting various linguistic features to learn the implicit discourse relation classifiers based on the training data collected (Wellner, Pustejovsky and Havasi, 2006; Pitler, Louis and Nenkova, 2009; Lin, Kan and Ng 2009; Wang, Su and Tan, 2010). Most useful linguistic features (such as word pairs) are extracted from the local context, which is usually determined as **Argument 1 (Arg1,** the first sentence/clause**)** plus **Argument 2** (**Arg2**, the second sentence/clause). Like the other related work in the literature, in this paper, we focus on the recognition of local implicit discourse relations, i.e. only the two arguments are examined.

To collect training data, the state-of-the-art methods normally start from the artificial/real perspective and simply make use of the implicit relations either derived from explicit or manual annotations. Marcu and Echihabi (2002); Sporleder and Lascarides (2008) created artificial implicit relations as training data by removing discourse connectives from the explicit relation examples. The advantage of these methods is that a large number of (artificial) implicit relation examples could be used as training data, saving the labor extensive and time-consuming annotation work. However, the experimental results in Sporleder and Lascarides (2008) showed that training on a large artificial data set is not necessarily a good strategy. Lin, Kan and Ng (2009) also pointed out that an artificially implicit relation corpus may exhibit marked differences from a natively implicit one. Also surprising is the fact that the results were not as good as expected when the classifiers are trained by using the manually annotated real implicit relations, though better than the results based on the artificial implicit relations.

Then the following questions come to our minds. Do all the real natively implicit relation examples provide useful hints for training the classifiers? Is it a reasonable choice if the training data is over-restricted to the annotated implicit relation examples even when the quantity of these data is limited and their annotation demands a high cost? Can a part of, if not all of, the artificial implicit relations created from the explicit relations be picked out to train an implicit relation classifier? In short, can we obtain more effective training examples at less cost?

With the above consideration, we argue that an effective training set is composed of typical examples, which have distinct characteristics to signify their discourse relations. These typical examples, however, can be either the natively implicit relations or the created implicit relations with connectives removed from the explicit relations. Using the typical examples as training data,

an implicit relation classifier with higher discrimination power can be built according to the linguistic features in the two arguments.

We provide three *Comparison* relation examples from the Penn Discourse TreeBank (PDTB) v2.0 (Prasad et al., 2008) which is widely used in the research of relation recognition as follows to illustrate what the possible typical examples are like.

(1) **Arg 1**: 44 North Koreans oppose the plan,
Arg 2: (while) South Koreans, Japanese and Taiwanese accept it or are neutral.
(2) **Arg 1**: In such situations, you cannot write rules in advance.
Arg 2: you can only make sure the President takes the responsibility.
(3) **Arg 1**: Columbia Savings is a major holder of so-called junk bonds.
Arg 2: New federal legislation requires that all thrifts divest themselves of such speculative securities over a period of years.

Here, the first one is an artificial implicit relation with the connective (i.e. "*while*") deleted while the second and third examples are natively implicit. The first and second ones are possibly typical because they have distinguishable linguistic features (such as: oppose/accept, cannot /can) to verify their relations. In contrast, it is hard to find significant characteristics in the third one to determine its discourse relation. The trained implicit relation classifier would possibly suffer a decline in performance if a lot of examples like the third one are included in the training set.

Based on the analysis above, we for the first time propose to select training data for implicit discourse relation recognition from a new typical/atypical perspective other than from an artificial/real perspective. Identifying the typical examples from both artificially created and real implicit discourse relations is the focus of this work. Assuming that the typical examples of a discourse relation are usually connected through the similar features, Yarowsky's algorithm (1995), as one of the first bootstrapping algorithms, gives us the following inspiration: given a small set of *seed* typical discourse relation examples, more typical examples are added iteratively by identifying the significant features of the seed set. In this paper, a training data selection approach named single centroid clustering (SCC) is proposed to acquire the typical examples for each relation. With the typical examples in the training set, the task of implicit relation recognition is cast to a classification problem. The experimental results show that the training set selected in such a way can improve the performance of an implicit relation classifier.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes our framework of implicit relation recognition, and introduces the types of features involved. Section 4 proposes the single centroid clustering algorithm that selects the typical examples iteratively. Section 5 presents the experimental results. Section 6 concludes our work.

2 Related Work

2.1 Implicit discourse relation recognition

So far, the existing research which used statistical models to recognize implicit discourse relations mainly falls into two categories according to whether the data annotation is required.

One research line tried to use the large quantity of unannotated explicit relations as a training set, which are roughly identified by discourse connectives and then converted to artificial implicit relations through removing the discourse connectives. Among the pioneer work was the one

presented by Marcu and Echihabi (2002) who applied massive amounts of unannotated explicit relations and lexical features to train the Naïve Bayes classifier for both explicit and implicit discourse relation recognition. Following the same idea, Saito, Yamamoto and Sekine (2006) conducted the experiments with the combination of cross-argument word pairs and phrasal patterns as features on Japanese sentences. Blair-Goldensohn（2007）further extended the work of Marcu and Echihabi (2002) by involving syntactic filtering and topic segmentation. Another interesting work is that of Zhou et al. (2010), which predicted discourse connectives between arguments via a language model. Then the generated connectives plus other linguistic features were combined in a supervised framework to determine the implicit discourse relation.

However, Sporleder and Lascarides (2008) discovered that the models of Marcu and Echihabi (2002) did not perform well on implicit relations recognition with artificially created relations as training data and concluded that removing discourse markers may lead to a meaning shift in the examples. Sporleder and Lascarides (2008) promoted the other research line that used the human-annotated training data. The development of various discourse banks also made the use of human-annotated data feasible. Based on Rhetorical Structure Theory Discourse Treebank (RST-DT) (Carlson et al. 2001), Soricut and Marcu (2003) developed two probabilistic models to identify elementary discourse units and generate discourse trees at the sentence level. Further Hernault et al. (2010); Feng and Hirst (2012) explore various features for discourse tree building on RST-DT. With the Discourse Graphbank (Wolf and Gibson, 2005), Wellner et al.(2006) integrated multiple knowledge sources to produce syntactic and lexical semantic features, which were then used to automatically identify and classify explicit and implicit discourse relations. Especially after the release of the second version of the PDTB v2.0 (Prasad et al., 2008), more research began to take the advantage of the annotated implicit relations for training purpose and were dedicated to exploiting various linguistic features in the supervised framework (Pitler, Louis and Nenkova, 2009; Lin, Kan and Ng, 2009; Wang, Su and Tan, 2010). Lin, Kan and Ng (2009) conducted a thorough performance analysis for four classes of features including contextual relations, constituent parse features, dependency parse features and cross-argument lexical pairs, while Pitler et al. (2009) applied several linguistically informed features, such as word polarity, verb classes, and word pairs. Wang, Su and Tan (2010) adopted the tree kernel approach to mine more structure information and got better results. These efforts of feature selection have achieved better performance though not that satisfying. The quality of training data are partly responsible for the difficulty of improving the performance of implicit relation recognition.

To better recognize the implicit discourse relations, we propose to review the annotated discourse corpora available at hand, identify and choose typical relation examples as training data for supervised learning. To the best of our knowledge, this paper is the first time to re-think the training data and implicit relation recognition from a novel perspective.

2.2 Rhetoric Discourse Treebank and Penn Discourse Treebank

As for the available discourse corpora, due to the space limitation we mainly introduce the two widely used discourse corpora - the Penn Discourse TreeBank (PDTB) and Rhetorical Structure Theory Discourse Treebank (RST-DT), which provide a common platform for researchers to develop discourse-centric systems.

The PDTB focuses on encoding discourse relations with the discourse connectives, adopting a lexically grounded approach for the annotation. For each pair of adjacent sentences within the same paragraph, annotators selected the explicit or implicit discourse connective which best

expressed the relation between the sentences. Then, the annotations can be seen as being of a predicate-argument structure, where a discourse connective is treated as a predicate taking a pair of adjacent sentences as its arguments. Thus, this discourse connective grounded approach exposes a clearly defined level of discourse structure. In PDTB, a hierarchy of relation tags is provided for the relation annotation. In our experiments, we only use the top level of the annotations, which is composed of four major relation classes: *Temporal*, *Contingency*, *Comparison* and *Expansion*. These four core relations allow us to be theory-neutral, since they are almost included in all discourse theories, sometimes under different names.

RST-DT is manually annotated under the Rhetoric Structure Theory framework (Mann and Thompson, 1988). In this corpus the rhetoric relations are labelled hierarchically between non-overlapping adjacent text spans which range from elementary discourse units (EDU, the minimal building blocks of a discourse tree) to paragraph. A total of 110 different relations were used for the tagging of the RST corpus (RST-DT, 2002). The final inventory of the relations is data driven and can be partitioned into 18 classes, from which we still select four classes including *Temporal*, *Contrast*, *Cause*, and *Background* to verify our method. These four relations spanning over individual sentences are collected to keep consistent with the discourse relations from PDTB.

So far, most of the previous works experimented on one corpus only. With the aim to verify the portability of our methods, we examine two corpora in this paper.

3 The Learning Framework for Implicit Discourse Relation Recognition

In this paper, the problem of implicit relation recognition is approached in the supervised learning framework. Figure 1 illustrates the architecture of our system.

The first and most important step is to collect the training data. As stated in Section 1, on the one hand, not all the annotated implicit relations contain significant features to distinguish themselves from the other relation types. On the other hand, we expect to pick out the suitable examples of the artificial implicit relations and strengthen their influence on the training process. We argue that the examples suitable to be training data are generally the typical ones having distinct linguistic features to signify their discourse relations, yet they can be of real implicit relations (denoted as *IM* data) or artificially implicit relations with connectives removed from explicit relations (denoted as *EX* data).

To select typical examples, for each discourse relation type the original artificial/real partition (denoted as EX_i/IM_i) is converted to a novel typical/atypical partition (denoted as A_i/B_i), which is obtained automatically by the proposed *single centroid clustering* (SCC) algorithm. Starting from an initial seed set, SCC iteratively refines typical examples and removes atypical examples if necessary. This algorithm is detailed in Section 4.

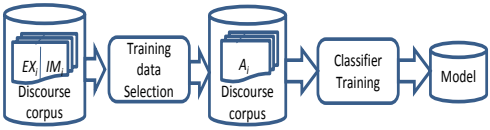

FIGURE 1 – System architecture for implicit discourse relation recognition

Assume there are n discourse relations. Let $Y=\{R_0, R_1, \ldots, R_n\}$ where R_i represents the ith typical relation type and R_0 denotes the atypical case. After the conversion from artificial/real partition to

typical/atypical partition, we get the $A_i/B_i(1 \le i \le n)$ and assign the relation label $R_i(1 \le i \le n)$ to each typical example in the set A_i. Each example in the union $A_0 = \cup B_i(1 \le i \le n)$ is labelled as R_0. Then the set of ordered pairs $<A_i, R_i>$ $(0 \le i \le n)$ can be used to train an implicit relation classifier for labelling $R_i(1 \le i \le n)$. Both clustering and classification require representing the annotated argument pairs with feature vectors. We introduce the feature selection in subsection 3.1.

3.1 Feature Selection

Various linguistic features have been experimented for recognizing implicit discourse relations in previous studies (Marcu and Echihabi, 2002; Pitler, Louis and Nenkova, 2009; Lin et al., 2009). Learning from them, we consider the following 7 types of features.

Polarity: The polarity of each sentiment word is tagged as positive, negative or neutral according to Multi-perspective Question Answering Opinion Corpus (Wilson et al., 2005). Note that the sentiment words preceded by negated words would be assigned an opposite tag. For example, "good" would be assigned as positive while “not good” is negative. Negated neutral is ignored. The occurrence of negative, positive and neutral polarities in each argument and their cross product are used as features.

Inquirer tags: General Inquirer lexicon (Stone et al., 1966) divides each word into fine-grained semantic categories described by the inquirer tags. From all the categories, we select 21 pairs of complementary categories, such as: *Rise* versus *Fall*, or *Pleasure* versus *Pain*, etc. The occurrence of each complementary category pair in the two arguments are used as features.

Modality: The presence of modal words including their various tenses and abbreviations in both arguments and their cross product are used as features.

SameWord: This type of feature represents whether a noun or a verb simultaneously occurs in both arguments. The intuition of using this feature is similar to that of the ***Verbs*** feature in (Pitler et al., 2009), for indicating the semantic association of the two arguments.

FirstLastFirst3: The first word, the last word, the first three words of each argument, the pair of the two first words and the pair of the two last words in the two arguments are used as features.

CrossWordPairs: The words in each argument compose one set. This type of features indicates the word pairs from the cross product of the two sets.

IntraWordPairs: The word pairs that occur in the same argument.

Since the length of the two arguments is relatively short, it is quite common that a feature is observed only once if it is present. Hence each feature is assigned a binary value to indicate whether it is present or absent. Assuming d features are extracted, each example is represented with a d-dimension binary feature vector.

4 Single Centroid Clustering for Training Example Selection

4.1 Overview

A good training set usually exhibits the property that most of its items have distinct features to differentiate the instances in the different classes. To precisely classify implicit discourse relations, the typical examples which have significant linguistic features except discourse connectives for identifying their relations are fit to be included in the training set. In this section,

we introduce the Single Centroid Clustering (SCC) algorithm which picks out the typical examples for each discourse relation from both *EX* and *IM* data.

Algorithm 1: Single-Centroid Clustering algorithm

Input: For relation i, artificial implicit relation set EX_i, real implicit relation set IM_i,
Output: Typical relation example set A_i, Atypical relation example set B_i

1. Initialize A_i: A_i= seed set of typical examples;
2. $B_i = EX_i \cup IM_i - A_i$
3. Compute the centroid CA_i for A_i
4. While stopping criterion has not been met
5. For each example e_j in A_i:
6. If dist $(e_j, CA_i) > T_{dis}^{(i)}$:
7. $A_i = A_i - \{e_j\}$; $B_i = B_i \cup \{e_j\}$
8. For each example e_j in B_i:
9. If dist$(e_j, CA_i) <= T_{dis}^{(i)}$:
10. $A_i = A_i \cup \{e_j\}$; $B_i = B_i - \{e_j\}$
11. Compute the centroid CA_i for A_i
12. End While

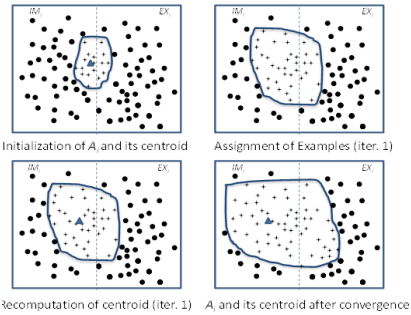

FIGURE 2 – Illustration of the Single Centroid Clustering algorithm

The principle underlying SCC is similar to that of the Yarowsky algorithm (1995), which has been successfully applied to the Word Sense Disambiguation (WSD). Yarowsky augmented the seed sets of each sense based on two powerful constraints, namely one-sense-per-collocation and one-sense-per-discourse. In our SCC algorithm, the features introduced in Section 3.1 are used to obtain the constraints of augmenting the seed sets and pick out those typical examples for each discourse relation. The SCC algorithm, as shown in Algorithm 1, consists of two loops. The "outer loop" can be regarded as a supervised learning process. In particular, based on the current available typical examples, SCC computes for each relation the centroid that judges which features are significant. The "inner loop" uses the current centroid of a relation to re-assign all the examples of the relation as either typical or atypical.

Figure 2 illustrates a snapshot of SCC on relation R_i, with dots and crosses representing the data in the A_i and B_i sets respectively. The closed curve in the left-top graph represents the seed set of typical examples. The closed curves in the other three graphs represent the intermediate and final results of the typical examples sets. The solid triangles in the middle of the closed curves denote the centroids computed based on typical examples. When SCC reaches its stable state, the final typical example set is passed to the classification models as training data. Take the three sentence pairs in Section 1 for example, the ideal output from SCC should include the first and the second examples in the typical set of the Comparison relation.

4.2 Implementation Details

4.2.1 Seed Set Construction

For each relation $R_i(1 \le i \le n)$, we can identify a relatively small number of typical examples as the seed set either manually or automatically. Similar to the Yarowsky algorithm (1995), to avoid the laborious procedure, through observation we manually lay down some simple rules to identify the distinct features for each relation from the 7 feature types and then select those containing the distinct features from the corresponding relation examples to compose of the seed set. The rules for identifying distinct features are illustrated in Table 1. Taking the *Comparison* relation for example, rule (1) identify the features of "Arg 1 is positive and Arg 2 is negative" and "Arg 1 is negative and Arg 2 is positive" which are from the **Polarity** feature type. Rule (2) can identify the features which are related to the words *seldom*, *back*, etc. according to the feature types of **FirstLastFirst3**, **CrossWordPairs**, and **IntraWordPairs**. Other strategies of selecting typical example seed set and the experimental comparisons are provided in Subsection 5.3.

Class	Description of Rules
Comparison	(1) A pair of opposite polarity tags is identified respectively in Arg 1 and Arg 2. (2) Arg 1 or Arg 2 contains the words including *seldom, back, yet, only.*
Contingency	(1) Opposite polarity tags are identified respectively in Arg 1 and Arg 2. (2) Arg 1 or Arg 2 contains the words including *draw, as, result.*
Temporal	Arg 1 or Arg 2 contains the words including *following, last, first, second.*
Expansion	Arg 1 and Arg 2 contain the same noun words or verb words.

TABLE 1 – Rules for selecting the seed set of typical examples.

4.2.2 Centroid Computation

A_i can be seen as the iteratively refined typical set. Suppose A_i is composed of $|A_i|$ examples $\{e_1^{(i)}, e_2^{(i)}, \ldots, e_{|A_i|}^{(i)}\}$, each example $e_j^{(i)}$ $(1 \le j \le |A_i|)$ is represented by a d-dimension feature vector $(e_{j1}^{(i)}, e_{j2}^{(i)}, \ldots e_{jd}^{(i)})$. In the d-dimensional Boolean space, the centroid CA_i is also represented by a d-dimension binary feature vector $(c_1^{(i)}, c_2^{(i)}, \ldots, c_d^{(i)})$, where $c_k^{(i)}$ is the value in the k^{th} dimension. We define $c_k^{(i)}$ as:

$$c_k^{(i)} = \begin{cases} 1, & \frac{\sum_j e_{jk}^{(i)}}{|A_i|} > T_c^{(i)} \\ 0, & otherwise \end{cases} \quad (1)$$

where $T_c^{(i)}$ is the percentage threshold corresponding to R_i. $c_k^{(i)}$ is assigned to 1 if the k^{th} feature occurs more than a certain percentage (i.e. $T_c^{(i)}$) of the examples that belong to the typical set A_i.

In this way, the centroid values actually reflect which features are significant to the corresponding discourse relation. Normally, centroid is used to compute the "average" of all objects in a certain space, and it should be noted that the computation of *centroid* in a Boolean space here does not strictly observe the "average" form.

4.2.3 Distance Metric

For each relation R_i, we exclude atypical examples from A_i or select typical ones into A_i by computing the distance between discourse relation examples and the centroid of CA_i. Assuming the example e is represented by the feature vector $(e_1, e_2, \ldots, e_d)$, the distance between e and CA_i is defined as follows.

$$dist(e, CA_i) = \sum_k w_k \cdot |c_k^{(i)} - e_k| \quad (2)$$

$$w_k = \begin{cases} \frac{|A_i| \cdot T_c - \sum_k e_{jk}^{(i)}}{|A_i|}, c_k^{(i)} - e_k = -1 \\ \frac{\sum_k e_{jk}^{(i)}}{|A_i|}, \ otherwise \end{cases} \quad (3)$$

where $|c_k^{(i)} - e_k|$ reflects whether the example e has a different value from CA_i in the k-th dimension and $w_k (1 \leq k \leq d)$ is used to measure the influence of the difference in the k-th dimension on the distance between e and CA_i. Here, w_k is determined according to the frequency of the k-th feature occurring in all examples of a discourse relation. The distance between an example e and the centroid CA_i denotes the representativeness or to say the typicality of the example e to the relation R_i. The smaller the distance value of an example, the more typical the example is.

A distance threshold $T_{dis}^{(i)}$ is set to control which examples should be selected into the typical set of R_i. The examples with distance less than $T_{dis}^{(i)}$ are possibly re-assigned to the typical set A_i. $T_{dis}^{(i)}$ is defined depending on the maximum distance and the minimum distance between the examples and the centroid CA_i, i.e.,

$$T_{dis}^{(i)} = min_e dist(e, CA_i) + p^{(i)} (\max_e dist(e, CA_i) - \min_e dist(e, CA_i)) \quad (4)$$

where $p^{(i)}$ is a control parameter within the interval (0,1) for R_i. If $p^{(i)}$ is set 0, $T_{dis}^{(i)}$ equals to the minimum distance, meaning that no examples can be included into the typical set. On the other extreme, if $p^{(i)}$ is 1, $T_{dis}^{(i)}$ equals to the maximum distance, it allows all the examples to be selected. The value of $p^{(i)}$ is also tuned to assure that typical examples can be well selected in each iteration.

5 Experiments and Evaluation

5.1 Experiment Set-up

The experiments and evaluations are conducted on the PDTB and RST-DT corpus, which contains 2519 and 385 Wall Street Journal articles respectively. PDTB is mainly used to evaluate and analyse recognition performance of our methods. RST-DT is used to verify the portability.

Following the work of Pitler, Louis and Nenkova (2009), the sections 2-20 of PDTB are used for training, the sections 0-1 for development and the sections 21-22 for test. As for the discourse relations, we adopt the top level of PDTB's annotations, which is composed of four major relation classes: *Temporal*, *Contingency*, *Comparison* and *Expansion*. Though PDTB allows each

sentence pair to be annotated with more than one relation, we only extract the first relation labelled for each sentence pair here. Table 2 shows the number of each relations in PDTB.

Class	Training		Test	Develop.
	EX	*IM*	Implicit	Implicit
Comparison	4209	1894	146	191
Contingency	2505	3281	277	287
Temporal	2633	665	67	54
Expansion	4770	6792	556	651
Total	14117	12632	1046	1183

TABLE 2 – Discourse relation distribution in PDTB.

According to the 7 types of features introduced in Section 3.1, in total 4022 features are extracted. Then each sentence pair is represented as a 4022-dimension binary feature vector. The two classifiers, i.e., the Naïve Bayes and Decision Tree classifiers are implemented with MALLET[1]. Two metrics, i.e., accuracy and F_1 measure, are used to evaluate the performance:

$$\text{Acc} = \frac{TruePositive + TrueNegative}{All} \quad \text{and} \quad F_1(R_i) = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

where *precision* and *recall* are two most common criteria to evaluate information retrieval and information extraction systems.

Four sets of experiments are designed (1) to tune the two thresholds $T_c^{(i)}$ and $T_{dis}^{(i)}$ in SCC; (2) to compare different strategies of selecting seed sets for SCC; (3) to compare the performance of various training sets on different classifiers; (4) to verify the portability of our methods.

5.2 Threshold Tuning in SCC

SCC aims at selecting typical examples for training discourse classifiers. Since it is difficult to directly evaluate the quality of a training set, we evaluate the training set outputted by SCC via the classification performance of a Decision Tree classifier. For each discourse relation R_i, SCC involves two main thresholds. $T_c^{(i)}$ determines which features are significant to the relation R_i, and $T_{dis}^{(i)}$ defines the borderline between the typical examples and the atypical ones. It is hard to find a global optimized solution for the combination of these two factors. So we apply a gradient search strategy. As in formula (4), $p^{(i)}$ is the only determining factor of $T_{dis}^{(i)}$. At first we set $p^{(i)}$ the value of 0.5, and different values of $T_c^{(i)}$ ranging from 0.05 to 0.35 are examined. Then, given that $T_c^{(i)}$ is set to the value with the best performance, we conduct experiments to find an appropriate value for $p^{(i)}$.

We ran four binary classifiers to distinguish each discourse relation (*Comp.*, *Cont.*, *Temp.*, and *Expa.* for short) from the others. For each relation, we include equal number of positive and negative examples in the training data. The positive examples are selected from the typical set of the relation while the negative examples are randomly chosen from the atypical set of the same relation or the other discourse relations. We use all the 1183 implicit relations in the development set, which is representative of the natural distribution of implicit discourse relations. Table 3 lists the F_1 and accuracy (within parentheses) of the implicit relation classifiers.

[1] www.mallet.cs.umass.edu.

T_c	*Comp*. vs. other	*Cont*. vs. other	*Temp*. vs. other	*Expa*. vs. other
0.05	23.2 (54.2)	39.0 (24.3)	12.8 (54.5)	0 (45.0)
0.10	23.9 (40.4)	41.9 (26.5)	12.8 (46.4)	64.4 (53.6)
0.15	**27.9 (39.4)**	38.6 (24.6)	12.5 (43.6)	66.2 (54.6)
0.20	27.6 (24.1)	39.9 (37.4)	12.4 (47.1)	68.3 (55.4)
0.25	17.4 (75.1)	42.1 (29.1)	**13.4 (45.5)**	**71.0 (55.0)**
0.30	18.4 (73.0)	**42.3 (28.3)**	11.7 (45.2)	55.2 (50.0)
0.35	17.0 (74.9)	39.1 (24.8)	12.6 (43.0)	55.2 (50.0)

TABLE 3 – F_1 (Acc) with varying $T_c^{(i)}$ values (p=0.5).

Table 3 shows that the value of $T_c^{(i)}$ directly influences the quality of the generated training set. When $T_c^{(i)}$ is assigned a smaller value, more features will satisfy the percentage requirement. That means more features will be reflected in the centroid and it will cause the distance between an example and the centroid is closer to one another. Then when $T_{dis}^{(i)}$ is fixed, more examples will enter into the typical set. Oppositely, when $T_c^{(i)}$ is assigned a larger value, it is more difficult for a feature to satisfy the percentage requirement. Then less number of features is reflected in the centroid. Notice that in general cases when the value of $T_c^{(i)}$ is larger than 0.35, the generated centroid closely approaches to the zero vector and thus does not work in the typical example selection. According to the best F_1 of each relation, we set the $T_c^{(i)}$ values to **0.15**, **0.3**, **0.25** and **0.25** for *Comp*., *Cont*., *Temp*., and *Expa*. respectively.

$p^{(i)}$	*Comp*. vs. other ($T_c^{(i)}$=0.15)	Cont. vs. other ($T_c^{(i)}$=0.3)	Temp. vs. other ($T_c^{(i)}$=0.25)	Expa. vs. other ($T_c^{(i)}$=0.25)
0.1	23.1(42.0)	0(75.7)	6.0(37.2)	0(45.0)
0.2	25.6(45.5)	0(75.7)	8.8(38.1)	0(45.0)
0.3	24.3(66.4)	29.8(58.7)	9.6(52.0)	0(45.0)
0.4	26.0(38.0)	39.0(24.3)	13.0(28.6)	0(45.0)
0.5	**27.9(39.4)**	**42.3(28.3)**	**13.4(45.5)**	**71.0(55.0)**
0.6	23.4(74.0)	39.0(24.3)	13.7(44.9)	62.5(52.3)
0.7	1.8(81.9)	39.0(24.3)	13.3(42.8)	57.8(54.1)
0.8	1.8(81.9)	38.9(25.1)	11.7(35.5)	0(45.0)
0.9	1.8(81.9)	38.1(24.5)	11.6(37.5)	0(45.0)

TABLE 4 – F_1 (Acc) with varying $p^{(i)}$ values ($T_c^{(i)}$ is fixed).

Next, with the tuned $T_c^{(i)}$ values, we inspect the performance of SCC with different $T_{dis}^{(i)}$ by tuning the value of $p^{(i)}$. Table 4 illustrates that almost all the classification reach their best performance at around $p^{(i)}$=0.5 where the threshold is the average of the minimum and maximum distances of the examples to the corresponding centroid. Then, in the following experiments, we set the $T_c^{(i)}$ values to **0.15**, **0.3**, **0.25** and **0.25** for *Comp*., *Cont*., *Temp*., and *Expa*., and all values of $p^{(i)}$ to 0.5.

At the same time, we observe the constituents of the best training data set generated by SCC for each relation. Table 5 illustrates the distributions of the final training set. From this table we can see that both the *IM* examples and *EX* examples contribute to the final typical example sets which is composed of 6753 artificial examples and 7816 real ones , According to Table 2 and Table 5, about 61.8 percent (7816/12632) of the *IM* examples and 47.8 percent (6753/14117) of the *EX* examples are typical. In the cases where the explicit discourse markers are absent, normally

richer linguistic features are involved to indicate the implicit discourse relations. For this reason, the real implicit examples tend to be typical.

	From EX_i	From IM_i	Total
$A_{Comp.}$	1293	852	2145
$A_{Cont.}$	1717	1418	3135
A_{Temp}	2090	404	2494
$A_{Expa.}$	1653	5142	6795
Total	6753	7816	14569

TABLE 5 – Constituents of the final typical sets.

5.3 Influence of Initial Seed Sets

The SCC algorithm begins with a seed set of typical examples that are picked out from the training data according to the manually summarized rules (denoted as the manual strategy) in section 4.2.1. The seed sets are generally composed of 1-5% of the corresponding relations.

stragegy	*Comp.* vs. other	*Cont.* vs. other	*Temp.* vs. other	*Expa.* vs. other
Manual	**27.9(39.4)**	**42.3(28.3)**	**13.7(44.9)**	**71.0(55.0)**
IM_seed	22.5(57.6)	39.8(47.4)	9.5(48.5)	50.9(47.4)
*EX*_seed	20.2(62.6)	39.0(24.3)	7.9(45.0)	55.2(50.0)
Random	19.1(75.7)	37.8(29.8)	8.0(27.9)	53.6(45.2)

TABLE 6 – F_1 (Acc) with different seed sets on Dev. Data.

stragegy	*Comp.* vs. other	*Cont.* vs. other	*Temp.* vs. other	*Expa.* vs. other
Manual	**28.5(62.0)**	**48.5(49.4)**	**14.7(69.0)**	**71.1(57.3)**
IM_seed	26.4(60.7)	41.9(26.5)	12.0(35.8)	52.6(49.2)
*EX*_seed	21.2(63.0)	41.9(35.4)	11.7(52.4)	54.6(50.1)
Random	22.2(47.1)	36.3(48.8)	11.1(40.2)	52.6(49.2)

TABLE 7 – F_1 (Acc) with different seed sets on Test Data.

For comparison purpose, we also examine the other three automatic seed set selection strategies on both development and test data. The results are shown in Table 6 and Table 7. We select the *IM* and *EX* data as seed set separately, denoted as *IM_seed* and *EX_seed* strategy respectively. With the Random strategy, we randomly select 10% of examples from the *EX* and *IM* data as the seed set for each relation. Both Table 6 and Table 7 show the superiority of the manual strategy over the other three. SCC to some extent is sensitive to the initialization of the typical set and could achieve a better performance with a better seed set of typical examples.

5.4 Evaluation of Implicit Relation Classifiers

We build four binary classifiers (*Comp.* vs Other, *Cont.* vs Other, *Temp.* vs Other, and *Expa.* vs Other) for relation labelling, and implement a 4-way classifier directly using the typical examples. All the 1046 implicit relations in the test data are used to compare our algorithm with the others.

Table 8 summarizes the performance implemented by Decision Tree (DT) and Naïve Bayes (NB) classifiers trained on different training sets in comparison with the state-of-the-art performance presented in Pitler et al. (2009), which solely uses the *IM* data to examine the influence of several linguistic features on implicit relation prediction. The second and third rows respectively show

Pitler's best results using single feature (Pitler-1) and combined features (Pitler-2), which are evaluated by a Naïve Bayes classifier. The IM_i, EX_i, EX_i+IM_i rows refer to our results of directly taking the *IM* data, the *EX* data, and both the *EX* and *IM* data as the training set respectively. Notice that all implementation of the IM_i method but feature selection is the same as Pitler's, though the performance of the IM_i method is far below Pitler's best results. This means feature selection is a key to promoting the performance.

		Comp. vs. Other	*Cont.* vs. Other	*Temp.* vs. Other	*Expa.* vs. Other	4-way
NB	Pitler-1	21.0(52.6)	36.7(62.4)	15.9(61.2)	71.3(59.2)	(65.4)
	Pitler-2	22.0(56.6)	47.1(67.3)	16.8(63.5)	76.4(63.6)	--
	IM_i	6.7(81.4)	41.9(28.0)	13.4(30.7)	44.4(51.9)	(51.3)
	EX_i	18.7(74.3)	40.1(27.6)	12.4(48.6)	8.2(46.6)	(34.1)
	EX_i+IM_i	14.0(76.5)	41.9(27.0)	12.7(44.8)	27.5(47.5)	(42.3)
	SCC	**24.3(58.3)**	**43.1(65.2)**	**18.0(92.2)**	**68.6(52.4)**	**(68.3)**
DT	IM_i	11.6(41.5)	38.7(40.5)	14.3(76.1)	38.8(44.7)	(53.5)
	EX_i	18.9(70.5)	41.9(26.5)	12.1(8.2)	0(46.8)	(42.6)
	EX_i+IM_i	14.0(76.5)	41.9(26.5)	9.0(67.3)	0(46.8)	(51.4)
	SCC	**28.5(62.0)**	**48.5(49.4)**	**14.7(69.0)**	**71.1(57.3)**	**(72.2)**

TABLE 8 – Performance comparison on PDTB.

SCC means using the training set which is composed of typical examples. Since the typical examples are picked out by SCC due to their distinct features, it is more suitable for the DT classifier to acquire the classifying rules according to the distinct features. That is why the performance of the DT classifier is better than that of the NB classifier in Table 8. The performance of both the DT and NB classifiers trained by typical examples are comparable to Pitler-1 and Pitler-2, though feature selection is not concerned in our systems. This table also shows that using typical examples as training data is more effective than using either IM_i, EX_i, or both IM_i and EX_i data as training set. For detecting the comparison relation with the DT classifier, the training set output by SCC significantly outperforms IM_i, by as much as about 17% absolute improvement in F_1-scores (i.e., 28.5 vs. 11.6). It is also observed that the performance of using IM_i as training set is comparable to that of using EX_i. This conforms to our assumption that typical examples contributes to the classification performance, while the final typical example set is composed of almost the same percent of the *IM* data and *EX* data according to Table 5.

According to the typical/atypical distribution in the training data, the test data should be composed of about 61.8% of typical ones and 38.2% of atypical ones. Since we do not preprocess the test data, the typical examples and the atypical ones in the test data are identified for their relations simultaneously. We observe the 4-way classification results with the DT classifier and find that most examples correctly identified are typical while the wrongly identified examples are usually atypical. For example, the third example in Section 1 is identified as *Expansion*.

5.5 Evaluation of Portability

To verify the portability of our method on RST-DT, we divide the whole RST-DT data into 347 training articles and 38 test articles. Different from PDTB, RST-DT includes about 18 relation types (RST-DT, 2002). To avoid data sparseness, we choose 4 relations that include a sufficient amount of examples. They are *Temporal*, *Contrast*, *Cause* and *Background*, and to some extent they are consistent with the 4 discourse relation types of PDTB. At the same time, we collect all

the 4 discourse relations spanning over individual sentences. Table 9 illustrates the relation distribution. For the 4 relations, we set $T_c^{(i)} = 0.25$ and $p^{(i)}=0.5$, SCC outputs the typical and atypical sets and their sizes are also given in the table.

Class	Training		Test	SCC	
	EXi	***IMi***	Implicit	A_i	B_i
Contrast	972	578	311	610	940
Background	701	677	330	660	718
Cause	304	785	535	846	243
Temporal	466	462	244	590	338

TABLE 9 – Relation distribution on RST.

Here, we evaluate the performance of SCC with the Decision Tree classifier. We compare it with the three baselines: real implicit examples (IM_i), artificial implicit examples (EX_i) or all the examples as training data (IM_i+EX_i). Table 10 shows that SCC can promote the performance with statistical significance (i.e., p-value[2]<0.1) on F_1. In addition, F_1 of *Contrast vs Other* (31.6) outperforms that of *Comparison vs Other* (28.5) on PDTB. It is the same for *Temporal*. According to our analysis, the reason is that the relations of RST-DT are fine-grained and it is relatively easy for SCC to obtain typical examples.

	Contrast vs. Other	*Background* vs. Other	*Cause* vs. Other	*Temporal* vs. Other
SCC	**31.6 (43.3)**	**38.3 (31.1)**	**54.8 (37.7)**	**31.2 (38.2)**
IM_i	27.6 (56.1)	34.8 (30.4)	35.6 (54.4)	29.2 (17.1)
EX_i	24.0 (64.9)	34.0 (41.5)	30.6 (57.1)	17.6 (67.0)
IM_i+EX_i	20.8(62.4)	34.9 (30.5)	32.2 (56.6)	27.2 (43.8)

TABLE 10 – F_1 (Acc) comparison on RST-DT.

Conclusions

In this paper, we for the first time present the typical/atypical perspective to select the most suitable training examples for implicit discourse relation recognition. A novel single centroid clustering algorithm is proposed to differentiate typical and atypical examples for each discourse relation. The experimental results show that the performance of the implicit relation classifiers with the typical examples selected as the training set are comparable to the best state-of-the-art methods on PDTB v2.0. In addition, the experiments on RST-DT show statistically significant improvements over the baselines and demonstrate the portability of our method. We will further explore more linguistic features and employ our approach on finer grained relation types. In SCC, we want to further investigate other distance formula. We also hope to explore the effective way to make use of the unlabelled discourse data.

Acknowledgments

The research work described in this paper has been partially supported by NSFC grants (No.61273278 and No.90920011), NSSFC grant (No: 10CYY023), National Key Technology R&D Program (No: 2011BAH10B04-03), and National High Technology R&D Program (No. 2012AA011101). We also thank the three anonymous reviewers for their helpful comments.

[2]Paired t-test is performed to compare the difference between SCC and *IM*, or between SCC and *IM+EX*. The *p*-values are 0.057 and 0.059 respectively.

References

Carlson, L., Marcu, D. and Okurowski, M.E. (2003). Building a discourse-tagged corpus in the framework of rhetorical structure theory. In Janvan Kuppelvelt and Ronnie Smith, editors, *Current Directions in Discourse and Dialogue*. Kluwer Academic Publishers.

Christiann, V. W. and Barnard., E. (2006). Data characteristics that determine classifier performance. *In Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, pp. 166–171, Parys, South Africa.

Feng, V. W. and Hirst, G., (2012). Text-level discourse parsing with rich linguistic features, *In Proc.of ACL'12*, pages 60-68.

Hernault, H., Prendinger, H., David, A., and Mitsuru I. (2010). HILDA: a discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1-33.

Lin, Z., Kan, M.–Y., and Ng, H. T. (2009). Recognizing implicit discourse relations in the Penn discourse treebank, *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: towards a functional theory of text organization. *Text*, 8(3):243-281.

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

Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A. and Joshi, A. (2008). Easily identifiable discourse relations. *In Proc. of the 22nd International Conference on Computational Linguistics (COLING08)*. pages 85-88.

Pitler, E., Louis, A. and Nenkova, A. (2009). Automatic sense prediction for implicit discourse relations in text, *In Proc. of the 47th ACL*. pages 683-691.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. and Webber, B. (2008). The Penn discourse treebank 2.0. *In Proc. of the 6th International Conference on Language Resources and Evaluation (LREC)*. Marrakech, Morocco.

RST_DT. (2002). RST Discourse Treebank. Linguistic Data Consortium, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07

Saito, M., Yamamoto, K. and Sekine, S. (2006). Using phrasal patterns to identify discourse relations. In Proc. of the HLTCNA Chapter of the ACL. pages 133-136.

Sasha B.-G. (2007). Long-Answer Question answering and rhetorical-semantic relations. Ph. D. thesis, Columbia Unversity.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. *In Proc. of HLT/NAACL 2003*. pages 149-156.

Sporleder, C. and Lascarides. A. (2008). Using automatically labelled examples to classify rhetorical relations: An Assessment. *Natural Language Engineering*, 14:369-416.

Wang, W., Su, J. and Tan C. L. (2010). Kernel based discourse relation recognition with temporal ordering information, *In Proc. of ACL'10*. pages 710-719.

Wilson, T., Wiebe, J. and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *In Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347-354.

Wellner, B., Pustejovsky, J., Havasi, C., Rumshisky, A. and Sauri, R. (2006). Classification of discourse coherence relations: an exploratory study using multiple knowledge sources. *In Proc. of the 7th SIGDIAL Workshop on Discourse and Dialogue*. pages 117-125.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods, *In Proc. of the 33rd annual meeting on Association for Computational Linguistics*, pages 189-196, Cambridge, Massachusetts.

Zhou, Z., Lan, M., Niu, Z. and Su, J. (2010). The effects of discourse connectives prediction on implicit discourse relation recognition, *In Proc. of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 139–146.