Identifying Cores of Semantic Classes in Unstructured Text with a Semi-supervised Learning Approach

Yun Niu and Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, ON, Canada M5S 3G4
yun,gh@cs.toronto.edu

Abstract

Cores of semantic classes in scenario descriptions can be extremely valuable in question-answering, information extraction, and document retrieval. We propose a semi-supervised learning approach to automatically identify and classify cores of semantic classes in unstructured text. We perform a case study on medical text. The results show that the selected features characterize the cluster structure of the data, and unlabeled data is effectively explored in the classification. Compared to a state-of-the-art supervised approach, the performance of the semi-supervised approach is much better when there is only a small amount of labeled data. The two are comparable when a large amount of labeled data is available.

Keywords

question answering, information extraction, transductive learning, named entity identification

1 Introduction

While the identification of named entities (NEs) in a text is an important component of many information retrieval and knowledge management tasks, including question answering and information extraction, its benefits are constrained by its coverage. Typically, it is limited to a relatively small set of classes, such as *person*, *time*, and *location*, for which instances can be recognized with reasonable confidence by straightforward methods with a minimal amount of context. However, in sophisticated applications, such as the non-factoid medical question answering that we consider in this paper, NEs are only a small fraction of the important semantic units discussed in documents or asked about by users. In fact, many semantic roles in scenarios and events that occur often in questions and documents do not contain NEs at all. Therefore, it is imperative to extend the idea of NE identification to other kinds of semantic units. In this paper, we propose an approach to detect a more diverse set of semantic units that goes beyond simple NEs.

Our targets are *cores* of semantic classes or roles in scenario descriptions. The semantics of a scenario is defined by the role that each participant plays in it and can be expressed by a frame structure, where each slot in the frame designates a semantic class. For example, a medical treatment scenario can have three semantic classes: the patient's problem P, the treatment or intervention I, and the clinical outcome O.[1] The slots in the corresponding frame may be filled with either *complete* or *partial* information. Consider the following example, where parentheses delimit each instance of a semantic class (a slot filler) and the labels P, I, O indicate its type:

> *Sentence:*
> Two systematic reviews in (people with AMI)$_P$ investigating the use of (calcium channel blockers)$_I$ found a (non-significant increase in mortality of about 4% and 6%)$_O$.
>
> *Complete slot fillers*:
> P: people with AMI
> I: calcium channel blockers
> O: a non-significant increase in mortality of about 4% and 6%
>
> *Partial slot fillers*:
> P: AMI
> I: calcium channel blockers
> O: mortality

The partial slot fillers in this example are the smallest fragments of the corresponding complete slot fillers that exhibit information rich enough for deriving a reasonably precise understanding of the scenario. We use the term *core* to refer to such a fragment of a slot filler. In this example, the cores of the patient's problem and the treatment are both NEs, whereas the core of the clinical outcome is not. Similarly, non-NE cores are common in other scenarios. For example, the *test method* in *diagnosis* scenarios, the *means* in a *shipping* event, and the *manner* in a *criticize* scenario may all have non-NE cores.

In a question answering system, keyword-based document retrieval is usually performed to find relevant documents that may contain the answer to a given question. Keywords in the retrieval are derived from the question. Cores of semantic classes can be extremely valuable in searching for such documents for complex question scenarios, as shown in this example.[2]

> *Question scenario:*
> A physician sees a 7-year-old child with asthma in her office. She is on flovent and ventolin currently and was recently discharged from hospital following

[1] Readers familiar with evidence-based medicine will recognize this as a simplification of the PICO representation for the formulation of a problem-centered query [21].

[2] The scenario is an example used in the usability testing in the EPoCare project at the University of Toronto.

> her fourth admission for asthma exacerbation. During the most recent admission, the dose of flovent was increased. Her mother is concerned about the impact of the additional dose of steroids on her daughter's growth. This is the question to which the physician wants to find the answer.

For a complex scenario description like this, the answer could be drowned in the large amount of irrelevant passages found by inappropriate keywords derived from the question. However, with the information given by cores of semantic classes, for example *P: asthma, I: steroids, O: growth*, the search can be much more effective.

Similarly, identifying cores of semantic classes in documents can facilitate the question/answer matching process. Some information relevant to the question is listed below, where boldface indicates a core:

> E1: A more recent systematic review (search date 1999) found three RCTs comparing the effects of **becolmetasone** and **non-steroidal medication** on linear **growth** in children with **asthma** (200 *μg* twice daily, duration up to maximum 54 weeks) suggesting a short term decrease in linear **growth** of -1.54 cm a year.

> E2: Two systematic reviews of studies with long term follow up and a subsequent long term RCT have found no evidence of **growth retardation** in **asthmatic children** treated with inhaled **steroids**.

The sentences here are from the book *Clinical Evidence* (CE) [3], which we are using as the base text in our project on natural-language question answering in evidence-based medicine [17]. The clinical outcomes mentioned in the evidence have very different phrasings — yet both are relevant to the question. The pieces of evidence describe two distinct outcomes. Missing either of the outcomes will lead to an incomplete answer for the physician. Here, the cores of semantic classes provide the only clue that both outcomes must be included in the answer, while complete description of semantic classes with more information could make the matching harder to find because of the different expressions of the outcomes.

In addition, semantics presented in cores of semantic classes can help filter out irrelevant information that cannot be identified by searching methods based on simple string overlaps. Consider these two questions:

> In patients with **myocardial infarction**, do β **blockers** reduce **mortality** and **recurrent myocardial infarction** without adverse effects?

> In someone with **hypertension** and **high cholesterol**, what management options will decrease his risk of **stroke** and **cardiac events**?

In the first question, the first occurrence of *myocardial infarction* is a disease but the second is part of the clinical outcome. In the second question, *stroke* is part of the clinical outcome rather than a disease to be treated as it usually is. Obviously, string matching cannot distinguish between the two cases. By identifying and classifying cores of semantic classes, the relations between these important semantic units in the scenarios are made very clear. Therefore, documents or passages that do not contain *myocardial infarction* or *stroke* as clinical outcomes can be discarded.

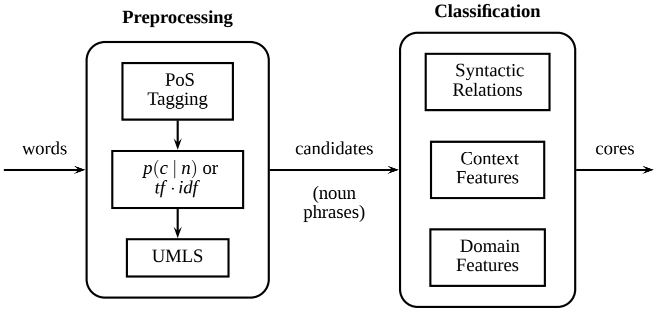

Fig. 1: *Architecture of the approach to core identification.*

Finally, cores of semantic classes in a scenario are connected to each other by the relations embedded in the frame structure. The frame of the treatment scenario contains a cause-effect relation: an intervention used to treat a problem results in a clinical outcome.

In the following sections, we propose a method to automatically identify and classify the cores of semantic classes according to their context in a sentence. We take the medical treatment scenario as an example, in which the goal is to identify cores of *treatments*, *problems*, and *clinical outcomes*. For ease of description, we will use the terms *intervention-core*, *disease-core*, and *outcome-core* to refer to the corresponding cores. We work at the sentence level, i.e., we identify cores in a sentence rather than a clause or paragraph. Two principles are followed in developing the method. First, complete slot fillers do not have to be extracted before core identification. Second, we aim to reduce the need for expensive manual annotation of training data by using a semi-supervised approach.

2 Architecture of the method

In our approach, we first collect candidates for the target cores from sentences under consideration. For each candidate, we classify it as one of the four classes: *intervention-core*, *disease-core*, *outcome-core*, or *other*. In the classification, a candidate will get a class label according to its context, its semantic types in the knowledge base Unified Medical Language System (UMLS), and the syntactic relations in which it participates. Two knowledge resources in UMLS — the Metathesaurus and the Semantic Network — are used. The Metathesaurus is the central vocabulary component of UMLS that contains information about biomedical and health-related concepts. Semantic types of concepts in the Metathesaurus are provided in the Semantic Network. Figure 1 shows the architecture of the approach.

3 Preprocessing

Our observation is that cores of the three types of slot fillers are usually nouns or noun phrases. In the preprocessing, all words in the data set are examined. The first two steps are to reduce noise, in which some of the words that are unlikely to be part of real cores are filtered out. Then, the rest are mapped to their corresponding concepts, and these concepts are candidates of target cores.

PoS tagging. Words that are not nouns are first removed from the candidate set. PoS tags are obtained by using

Brill's tagger [5].

Filtering out some "bad" nouns. This step is the second attempt to remove noise. Nouns that are unlikely to be part of real cores are considered to be "*bad*" candidates. Two different measures are considered to evaluate how *good* a noun is.

$tf \cdot idf$. This is the traditional measure of informativeness of a word with regard to a document. *Clinical Evidence* text is used to obtain the $tf \cdot idf$ value of a noun. 47 sections in *Clinical Evidence* are segmented to 143 files of about the same size. After the $tf \cdot idf$ value of a noun is calculated in each file, the highest value is taken as its final score. Nouns with $tf \cdot idf$ values lower than a threshold are removed from the candidate set. The threshold was set manually after observing the values of some nouns that frequently occur in the text.

Domain specificity. We calculate the probability $p(c|n)$, where c is the medical class, and n is a noun. This is the probability that a document is in the medical domain c given that it contains the noun n. Intuitively, *intervention-cores*, *disease-cores*, and *outcome-cores* are domain-specific, i.e., a document that contains them is very likely to be in the medical domain. For example, *morbidity*, *mortality*, *aspirin*, and *myocardial infarction* are very likely to occur in a medicine-related context. Therefore, we intend to retain highly medical domain-specific nouns in the candidate set. Using this measure, a noun is a better candidate if the corresponding probability is high. Text from two domains is needed in this measure: medical text, and non-medical text. In our experiment, we use the same 47 sections in *CE* as the medical class text. For the non-medical class, we use the Reuters collection, as it mainly consists of newswire stories. 1000 documents in the Reuters collection are randomly selected for the calculation. Nouns whose probability values are below a threshold (determined in the same manner as in the $tf \cdot idf$ measure) are filtered out.

Mapping to concepts. In many cases, nouns are part of noun phrases that are better candidates for cores. For example, the phrase *myocardial infarction* is a better candidate for an intervention-core than *infarction*. Therefore, we use the software MetaMap [2] to map a noun to its corresponding concept (which is often a noun phrase) in the Metathesaurus of UMLS. All the concepts form the set of candidates of cores to be classified.

4 Representing candidates using features

Given a set of candidates, the classification task is to identify several subsets; each corresponds to a type of slot filler, or a semantic class. We expect that candidates in the same semantic class will have similar behavior, characterized by syntactic relations, context information, and semantic types. All features are binary features, i.e., a feature takes value 1 if it is present; otherwise, it takes value 0.

4.1 Syntactic relations

Syntactic relations have been explored in grouping similar words [14] and words of the same sense in word sense disambiguation [12]. Lin [14] inferred that *tesguino* is similar to *beer*, *wine*, etc., i.e., it is a kind of drink, by comparing syntactic relations in which each word participates. Kohomban and Lee [12] determined the sense of a word by observing a subset of syntactic relations of the word. The hypothesis is that different instances of the same sense will have similar relations.

Sentence:
Thrombolysis reduces the risk of dependency, but increases the chance of death.
Candidates:
thrombolysis, dependency, death
Relations:
(thrombolysis subj–of increase), (thrombolysis subj–of reduce)
(dependency pcomp-n–of of)
(death pcomp-n–of of)

Fig. 2: *Example of dependency triples extracted from output of Minipar parser.*

We also need to group instances of the same semantic class. Such instances may participate in similar syntactic relations while those of different classes will have different relations. For example, intervention-cores often are subjects of sentences, while outcome-cores are often objects.

Candidates in our task are phrases, rather than words as in [14] and [12]. Thus, we consider all relations between a candidate noun phrase and other words in the sentence. To do that, we ignore relations between any two words in the phrase when extracting syntactic relations. Any relation between a word not in the phrase and a word in the phrase is extracted. We use the Minipar parser [13] to get the syntactic relations. In the feature construction, a relation triple containing two words and the grammatical relation between them is taken as a feature, as shown in Figure 2. The set of all distinct triples is the syntactic relation feature set in the classification.

4.2 Local context

The context of candidates is also important in distinguishing different classes. For example, a disease-core may often have *people with* in its left context. However, it is very unlikely that the phrase *people with mortality* (with an outcome-core) will occur in the text. We consider the two content words on both sides of a candidate. When extracting context features, all punctuation marks are removed except the sentence boundary. The window does not cross boundaries of sentences. We evaluated two representations of context: ordered and unordered. In the ordered case, local context to the left of the phrase is marked by *L-*, that to the right is marked by *R-*. Symbols *L-* and *R-* are used only to indicate the order of text. For the candidate *dependency* in Figure 2, the context features with order are: *L-reduces*, *L-risk*, *R-increases*, and *R-chance*. The context features without order are: *reduces*, *risk*, *increases*, and *chance*.

4.3 Domain features

Each candidate has a semantic type defined in UMLS. For example, the semantic type of *death* is **organism function** and that of *dependency* is **physical disability**. These semantic types are used as features in the classification.

Table 1: *Number of instances of cores in the whole data set.*

Intervention-core	501
Disease-core	153
Outcome-core	384
Total	1038

5 Data set and analysis

Two sections of *Clinical Evidence* were used in the experiments. A clinician labeled the text for intervention-cores and disease-cores. Complete clinical outcomes are also identified. Using this annotation as a basis, outcome-cores were labeled by the first author. The number of instances of each class is shown in Table 1.

In our approach, the design of the features is intended to group similar cores together. As a first step to verify how well the intention is captured by the features, we observe the geometric structure of the data.

In the analysis, candidates are derived using the domain specificity measure $p(c|n)$. Each candidate is represented by a vector of dimensionality D, where each dimension corresponds to a single feature. The feature set consists of syntactic features, ordered context, and semantic types. We map the high-dimensional data space to a low-dimensional space using the locally linear embedding (LLE) algorithm [20] for easy observation. LLE maps high-dimensional data into a single global coordinate system of low dimensionality by reconstructing each data point from its neighbors. The contribution of the neighbors, summarized by the reconstruction weights, captures intrinsic geometric properties of the data. Because such properties are independent of linear transformations that are needed to map the original high-dimensional coordinates of each neighborhood to the low-dimensional coordinates, they are equally valid in the low-dimensional space. In Figure 3, the data is mapped to a 3-dimensional space (the coordinate axes in the figure do not have specific meanings as they do not represent coordinates of real data). Candidates of the four classes (intervention-core, disease-core, outcome-core, and other) are represented by (red) stars, (blue) circles, (green) crosses, and (black) triangles, respectively. We can see that candidates in the same class are close to each other, and clusters of data points are observed in the figure.

6 The model of classification

On the basis of the feature design and data analysis, we choose a semi-supervised learning model developed by Zhu et al. [24] that explores the clustering structure of data in classification. The general hypothesis of the approach is that similar data points will have similar labels.

Let $x_1, \ldots, x_n$ be labeled and unlabeled data. In the model, a graph $G = (V, E)$ is constructed (it does not have to be fully connected), where the set of nodes V correspond to both labeled and unlabeled data points and E is the set of edges. The edge between two nodes i, j is weighted. Weight w_{ij} is assigned to agree with the hypothesis so that the edge between two nodes that are closer in the data space gets higher weight. This approach explores the cluster structure of data by propagating labels from labeled data points to unlabeled data points according to the weights on the edges. Zhu et al. formulate the intuitive label propagation approach as a problem of energy minimization in the framework of Gaussian random fields, where the Gaussian field is over a continuous state space, instead of over a discrete label set. The idea is to compute a *real-valued* function $f : V \rightarrow \mathcal{R}$ on graph G that minimizes the energy function $E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f(i) - f(j))^2$. The function $f = argmin_f E(f)$ determines the labels of unlabeled data points. This solution can be efficiently computed by direct matrix calculation even for multi-label classification, in which solutions are generally computationally expensive in other frameworks. It is referred to as "SEMI" in the following description.

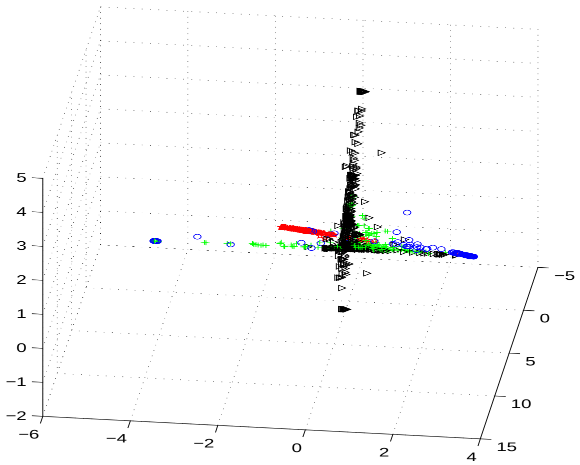

Fig. 3: *Manifold structure of data.*

Label propagation explores the similarity of labeled and unlabeled data points, and thus follows closely the cluster structure of the data in prediction. We expect it to perform reasonably well on our data set. We use the SemiL [10] implementation of SEMI in the experiment.[3]

7 Results and analysis

We first evaluate the performance of the semi-supervised model on different feature sets. Then, we compare the candidate sets obtained by using $tf \cdot idf$ with those obtained by evaluating domain specificity. Finally, we compare the semi-supervised model to a supervised approach.

In all these experiments, the data set contains all candidates of cores. Unless otherwise mentioned, the results reported are obtained using the candidate set derived by $p(c|n)$, the feature set of the combination of syntactic relations, ordered context, and semantic types, and the distance measure of cosine distance (as weights on the edges of the graph). The result of an experiment is the average of 20 runs. In each run, labeled data is randomly selected from the candidate set, and the rest is taken as unlabeled data whose labels need to be predicted. We make sure that all classes are present in labeled data; if any class is absent, we redo the sampling. The evaluation of the semantic classes is

[3] As our data is unbalanced, the parameter that handles unbalanced data set is turned on the experiment. Default values of other parameters are used unless otherwise mentioned.

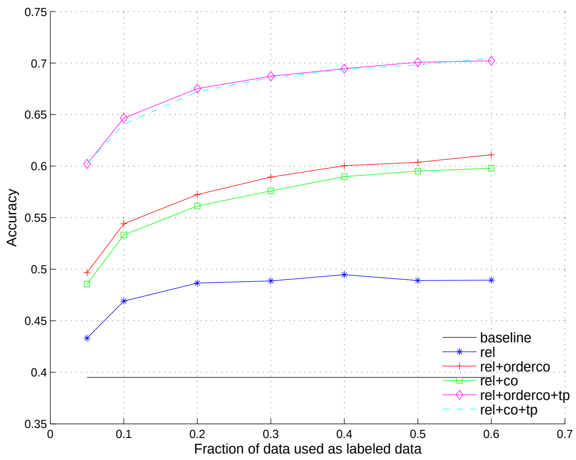

Fig. 4: *Classification results of candidates.*

very strict: a candidate is given credit if it gets the same label as given by the annotator and the tokens it contains are exactly the same as those marked by the annotator. Candidates that contain only some of the tokens matching the labels given by the annotators are treated as the *other* class.

7.1 Experiment 1: Evaluation of feature sets

This experiment evaluates different feature sets in the classification. As described in section 3, two different methods are used in the second step of preprocessing to pick up *good* candidates. Here, as our focus is on the feature set, for space reasons, we report only results on candidates selected by $p(c|n)$. (In section 7.2, we compare the two methods of selecting good candidates.)

Figure 4 shows the accuracy of classification using different combinations of the four feature sets: syntactic relations, ordered context, un-ordered context, and semantic types. A baseline is set by assigning labels to data points according to the prior knowledge of the distribution of the four classes, which has accuracy of 0.395. It is clear in the figure that incorporating additional kinds of features into the classification results in a large improvement in accuracy. Using only syntactic relations (*rel* in the figure) as features, the best accuracy is lower than 0.5. The addition of ordered context (*orderco*) or no-order context features (*co*) improves the accuracy by about 0.1. Adding semantic type features (*tp*) improves accuracy by a further 0.1. Combining all four kinds of features achieves the best performance. With only 5% of data as labeled data, the whole feature set achieves an accuracy of 0.6. Semantic types seems to be a very powerful feature set, as it substantially improves the performance on top of the combination of the other two kinds of features. Therefore, we took a closer look at the semantic type feature set by conducting the classification using only semantic types, but found that the result is even worse than using only syntactic relations. This observation reveals interesting relations between the feature sets. In the space defined by only one kind of features, data points may be close to each other, and hence hard to distinguish. Adding another kind sets apart data points in different classes toward a more separable position in the new space. This shows that every kind of feature is informative to the task. The feature sets characterize the candidates from different angles that are complementary in the task.

We also see that ordered context features are only slightly better than unordered features when semantic types are not considered. This difference is not observable at all when semantic type information is considered.

7.2 Experiment 2: Evaluation of candidate sets

In the second step of preprocessing, one of two methods can be used to filter out some *bad* nouns – using $tf \cdot idf$ value or the domain specificity. This experiment compares the two measures in the core identification task. A third option using neither of the measures (i.e., without filtering) is taken as the baseline. Table 2 shows numbers of instances remaining in the candidate set after preprocessing.

As shown in the table, there are much fewer instances in the *other* class in the sets derived by $tf \cdot idf$ and the probability measure as compared to those derived by the baseline, which shows that the two measures effectively remove some of the *bad* candidates of intervention-core, disease-core, and outcome-core. At the same time, a small number of cores are removed.[4] Compared to the baseline method, the probability measure keeps almost the same number of intervention-cores and disease-cores in the candidate set, while omitting some outcome-cores. This indicates that outcome-cores are less domain-specific than the other two. Compared to the $tf \cdot idf$ measure, more intervention-cores and outcome-cores are kept by the domain specificity measure, showing that the probability measuring the domain-specificity of a noun better characterizes the cores of the three semantic classes. The probability measure is also more robust than the $tf \cdot idf$ measure, which heavily relies on the content of the text from which it is calculated. For example, if an intervention is mentioned in many documents of a document set, its $tf \cdot idf$ value can be very low although it is a good candidate of intervention-core.

The precision, recall, and F-score of the classification shown in Table 3 confirms the above analysis. The probability measure gets substantially higher F-scores than the baseline for all the three classes that we are interested in, using different amounts of labeled data. In particular, the corresponding precision values are much higher than the baseline. Compared to $tf \cdot idf$, the performance of the domain specificity measure is much better on identifying intervention-cores, and slightly better on identifying outcome-cores, while the two are similar on identifying disease-cores.

7.3 Experiment 3: Comparison of the semi-supervised model and SVMs

In the semi-supervised model, labels propagate along high-density data trails, and settle down at low-density gaps. If the data has the desired structure, unlabeled data can be used to help learning. In contrast, a supervised approach only makes use of labeled data. This experiment compares SEMI to a state-of-the-art supervised approach; the goal is

[4] The first and third step in the preprocessing also results in missing cores in the candidate set. We roughly checked about one-third of the total real cores in the data set and found that 80% of lost cores occur because MetaMap either failed to find the concepts or it extracted more or fewer tokens than marked by the annotator. 10% of missing cores are caused by errors of the PoS tagger, and the rest occur because some cores are not nouns.

Table 2: *Number of candidates in different candidate sets.* Class 1: *intervention-core*, Class 2: *disease-core*, Class 3: *outcome-core*, Class 4: *other*

Measures	Class1	Class2	Class3	Class4
$tf \cdot idf$	243	108	194	785
$p(c\|n)$	298	106	209	801
baseline	303	108	236	1330

Table 3: *F-score of classification on different candidate sets.*

labeled data	1%	5%	10%	30%	60%
intervention-core:					
baseline	.53	.63	.66	.70	.72
$tf \cdot idf$	.51	.61	.64	.69	.71
$p(c\|n)$	**.57**	**.69**	**.72**	**.75**	**.77**
disease-core:					
baseline	.25	.36	.43	.48	.49
$tf \cdot idf$	**.29**	**.41**	.46	**.53**	**.55**
$p(c\|n)$	.27	**.41**	**.47**	**.53**	**.55**
outcome-core:					
baseline	.28	.41	.48	.53	.55
$tf \cdot idf$	.35	**.49**	.53	.59	.61
$p(c\|n)$	**.37**	**.49**	**.54**	**.60**	**.63**

Table 4: *F-score of classification using different models.*

labeled data	1%	5%	10%	30%	60%
intervention-core:					
semi	.57	.69	.72	.75	.77
SVM	.33	.60	.68	.74	.77
disease-core:					
semi	.27	.41	.47	.53	.55
SVM	.21	.38	.54	.62	.65
outcome-core:					
semi	.37	.49	.54	.60	.63
SVM	.07	.27	.44	.56	.62

to investigate how well unlabeled data contributes to the classification using the semi-supervised model. We compare the performance of SEMI to support-vector machines (SVMs) when different amounts of data are used as labeled data. We use OSU SVM [15] in the experiment.[5]

As shown in Table 4, when there is only a small amount of labeled data (less than 5% of the whole data set), which is often the case in real-world applications, SEMI achieves much better performance than SVMs in identifying all the three classes. For intervention-core and outcome-core, with 5% data as labeled data, SEMI outperforms SVMs with 10% data as labeled data. Similarly, SVMs need to have about three times the labeled data to gain the same performance achieved by SEMI using 10% data as labeled data. With less than 60% data as labeled data, the performance of SEMI is either superior to or comparable to SVMs for intervention-core and outcome-core. This shows that SEMI effectively exploits unlabeled data by following the manifold structure of the data. The promising results achieved by SEMI show the potential of exploring unlabeled data in classification.

8 Related work

The task of named entity (NE) identification, similar to the core-detection task, involves identifying words or word sequences in several classes, such as proper names (locations, persons, and organizations), monetary expressions, dates and times. NE identification has been an important research topic ever since it was defined in MUC [16]. In 2003, it was taken as the shared-task in CoNLL [22]. Most statistical approaches use supervised methods to address the problem [9, 6, 11]. Unsupervised approaches have also been tried in this task. Thelen and Riloff [23] explored a bootstrapping method to learn semantic lexicons of six categories: building, event, human, location, time, and weapon. Cucerzan and Yarowsky [8] also used a bootstrapping algorithm to learn contextual and morphological patterns iteratively. Collins and Singer [7] tested the performance of several unsupervised algorithms on the problem: modified bootstrapping (DL-CoTrain) motivated by co-training [4], an extended boosting algorithm (CoBoost), and the Expectation Maximization (EM) algorithm. The results showed that DL-CoTrain and CoBoost are superior to EM, while the two are almost the same.

Much effort in entity extraction in the biomedical domain has gene names as the target. Various supervised models including naive Bayes, support-vector machines, and hidden Markov models have been applied [1]. The work most related to our core-identification in the biomedical domain is that of Rosario and Hearst [19], which extracts *treatment* and *disease* from MEDLINE and examines seven relation types between them using generative models and a neural network. They claim that these models may be useful when only partially labeled data is available, although only supervised learning is conducted in the paper. The best F-score of identifying *treatment* and *disease* obtained by using the supervised method was .71. Another piece of work extracting similar semantic classes was that of Ray and Craven [18]. They report an F-score of about .32 for extracting *proteins* and *locations*, and an F-score of about .50 for *gene* and *disorder*.

9 Conclusion

We proposed a novel approach to automatically identify and classify cores of semantic classes in scenario descriptions. In the classification, a semi-supervised model that explores the clustering structure of the data was applied. Our experimental results show that syntactic relations, context, and semantic types are informative and complement features for this task. The features characterize the cluster structure of the data, and unlabeled data is effectively used. Compared to a state-of-the-art supervised approach, the performance of the semi-supervised approach is much better when there is only a small amount of labeled data, and performance of the two are comparable when larger amounts of labeled data are available.

Our approach does not require prior knowledge of semantic classes, and it effectively exploits unlabeled data.

[5] For the parameter that handles unbalanced data, we set it according to the prior knowledge of the class distribution and give larger weight to a class that contains fewer instances.

The promising results achieved show the potential of semi-supervised models that explore the clustering structure of data in tasks of grouping *similar* instances. This approach can be applied to other domains as well; the syntactic relation and context features can be constructed in the same way. For domains that do not have a knowledge base like UMLS, the WordNet hierarchy may be used to get features like semantic types. In this case, the level of generalization in WordNet needs to be investigated.

A difficulty of using this approach, however, is in detecting boundaries of the targets. A segmentation step that pre-processes the text is needed. In the next step of our work, we aim to investigate approaches that perform the segmentation precisely.

Acknowledgments. Our work is supported by a grant from the Natural Sciences and Engineering Research Council of Canada and grants from Bell University Laboratories at the University of Toronto. We thank Xiaodan Zhu, Jianhua Li, Sharon Straus, Suzanne Stevenson, Gerald Penn and John Mylopoulos for their helpful discussion and comments on this work.

References

[1] S. Ananiadou and J. Tsujii, editors. *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*. Association for Computational Linguistics (ACL), PA, USA, 2003.

[2] A. R. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *Proceedings of the American Medical Informatics Association Symposium*, pages 17–21, 2001.

[3] S. Barton, editor. *Clinical evidence*. BMJ Publishing Group, London, 2002.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[5] E. Brill. *A Corpus-Based Approach to Language Learning (PhD thesis)*. U of Pennsylvania, 1993.

[6] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of 7th Conference on Computational Natural Language Learning*, pages 160–163, 2003.

[7] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[8] S. Cucerzan and D. Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proc of the 1999 Joint SIGDAT Conf on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[9] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proc of 7th Conf on Computational Natural Language Learning*, pages 168–171, 2003.

[10] T.-M. Huang, V. Kecman, and I. Kopriva. *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, Berlin, Germany, 2006.

[11] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning. Named entity recognition with character-level models. In *Proc of 7th Conf on Computational Natural Language Learning*, pages 180–183, 2003.

[12] U. S. Kohomban and W. S. Lee. Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 34–41, 2005.

[13] D. Lin. Principar – an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 482–488, 1994.

[14] D. Lin. Automatic retrieval and clustering of similar words. In *Proc of the 17th International Conf on Computational Linguistics*, pages 768 – 774, 1998.

[15] J. Ma, Y. Zhao, S. Ahalt, and D. Eads. OSU SVM classifier Matlab toolbox. In *http://svm.sourceforge.net/docs/3.00/api/*, 2003.

[16] MUC. Message understanding conferences. In *http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html*, 1995.

[17] Y. Niu, X. Zhu, J. Li, and G. Hirst. Analysis of polarity information in medical text. In *Proceedings of Annual Symposium of American Medical Informatics Association*, pages 570–574, 2005.

[18] S. Ray and M. Craven. Representing sentence structure in hidden Markov models for information extraction. In *Proc 17th International Joint Conf on Artificial Intelligence*, pages 1273–1279, 2001.

[19] B. Rosario and M. A. Hearst. Classifying semantic relations in bioscience texts. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics*, pages 431–438, 2004.

[20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[21] D. L. Sackett, S. E. Straus, W. S. Richardson, W. Rosenberg, and R. B. Haynes. *Evidence-Based Medicine*. Harcourt, Edinburgh, 2000.

[22] E. F. T. K. Sang and F. D. Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL*, pages 142–147, 2003.

[23] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 214–221, 2002.

[24] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.