Acronym classification using feature combinations

Dana Dannélls
Department of Swedish Language
Göteborg University
SE-405 30 Gothenburg, Sweden
dana.dannells@svenska.gu.se

December 21, 2007

Abstract

This paper presents a supervised machine learning approach to the acronym-definition recognition problem. A task which is both difficult and crucial for many Natural Language Processing (NLP) applications. The emphasis is on the choice of particular information sources for the training experience and their effect on the learning system. We conducted an experiment which combines two machine learning approaches to classify acronym-definition pairs.

1 Introduction

Acronym-definition recognition is an universal problem which have been shown to be important at different levels, cross languages and domains (Zahariev, 2004). Applications such as information extraction and information retrieval are some areas where acronym-definition recognition is of vast importance. In recent developments, there have been a number of attempts to apply machine learning and statistical methods to address this problem, e.g., Nadeau and Turney (2005); Chang et al. (2002) showing these approaches outperform handcrafted rule systems.

One of the difficulties in recognizing acronym-definition pairs is their wide acronym-definition formation coverage, i.e., acronyms may appear in any length and may be realized in different surface forms with respect to their definition strings, especially in biomedical texts where the vocabulary is quickly expanding with new acronym-definition pairs. A way to approach this problem is simply by describing acronym-definition pairs as feature vectors and train a machine learning algorithm to classify them. Previous work on automatic acronym-definition recognition describe an extensive number of features which can be computed for different machine learning classification tasks (for a detailed overview see Dannélls (2006b)), some of which have shown promising results.

The choice of particular information sources for the training experience is one of the most important choices that a designer has to face when designing a machine learning system (Mitchell, 1997). The question is which particular information should the learning instances convey to allow flexibility when processing new inputs. This is the problem we focus on for the present work. In this paper we present an experiment which combines two machine learning approaches and enhances the information the training experience comprehends in order to try and improve upon previous results.

2 Memory-Based Learning

Memory-Based Learning (Daelemans, 1999) is an Instance-based Learning (IBL) paradigm known as *lazy learning* (Mitchell, 1997). Lazy learning approaches differ from statistical methods as they store previously encountered instances in memory and use them directly to process new inputs, rather than abstracting their statistical distribution. IBL is known to be able to learn exceptions in data and adapt to sub-regularities. The advantage of these learning algorithms is that they estimate the target function locally and differently for each new instance to be classified.

TiMBL[1] (Tilburg Memory-Based Learner) is a program which includes implementation of several memory-based learning algorithms. The program stores a representation of the training set explicitly in memory and classifies new cases by extrapolation from the most similar stored cases. TiMBL is optimized for fast classification by using several indexing techniques and heuristic approximations. It gives access to several memory-based learning algorithms and metrics, some of which are: Information-gain weighting for dealing with features of differing importance and modified value difference metric for making graded guesses of the match between two different symbolic values.

One way of using TiMBL is to apply a learning method to a dataset and analyze its output to extract information about the data. A particular problem can be compared and evaluated using different methods, this flexibility makes the program suitable for selecting the most appropriate method and representation for a certain learning task.

Machine learning problem

The are several ways to approach the acronym-definition recognition in terms of machine learning problem. In this work we are concerned with two approaches.

The first approach is by describing the acronym-definition recognition task as a binary classification problem. Given a set of features that describe an acronym-definition pair, made of an acronym A (a single token) and a definition D (a sequence of one or more consecutives tokens), determine whether the pair (A, D) is a positive instance.

[1] http://ilk.uvt.nl

The second approach is similar to part-of-speech, chunking and other natural language tagging tasks. Given a definition string (D), that is a sequence of letters, determine the appropriate action on each letter that corresponds to the letters of the acronym (A). From a probabilistic modeling point of view, the task is to find the sequence of actions $t_1 \ldots t_n$ that maximizes the probability $P(t_1 \ldots t_n | o)$, given the observation $o = o_1 \ldots o_n$, where observations are the letters in the definition and various types of features derived from them.

3 Data and methodology

Our data was extracted from the MEDLEX corpus (Kokkinakis, 2006) of Swedish medical texts. It consists of 671 acronym-definition pairs, of which 47 are negative examples. The data was partitioned into training (80%) and test (20%) sets, resulting in 537 training examples and 134 test examples. This is the same data and distribution that we experimented with and presented in Dannélls (2006a). The results reported there (with the best accuracy of 96.3% by IGTREE) are used as our baseline, upon which we try to improve by computing an observation probability for each acronym-definition pair and combine it with the original feature set which consists of ten features. A motivation for computing observation probabilities is that probabilistic approaches are known to have a greater coverage of syntactic constrictions and have been enormously successful in different areas such as information retrieval.

We utilized the MEMM-based tagger which was implemented by Tsuruoka et al. (2005). Tsuruoka et al. (2005) use a Maximum Entropy Markov Model (MEMM) to generate acronyms together with their probabilities from the letters in the definition string. Their tagger is based on eight features that describe the letters in the definition string, these are: uni-gram, bi-gram, tri-gram, uppercase, action history, definition length, letter sequence and distance sequence between the target letter and the beginning and tail of the word. In order to train the MEMM tagger we tagged each letter of the definition string with actions that correspond to the letter of acronym. In total we used five actions which are classified into the following classes: skip, upper, lower, space, and hyphen.

The Maximum-Entropy Markov Model was trained on 624 positive instances from the MEDLEX, using five-fold cross validation experiments.[2] The tagger achieved a coverage of 91,6% for the top ten Swedish acronym-definition candidates. Yielding very low probabilities for a few correct acronym-definition pairs, e.g. P(**b**udbärar**RNA** (mRNA)) = 0.001. To assign probabilities to the remaining 47 negative instances we tested the model on this small set, it coveraged 75.1%. The results showed there were a number of incorrect acronym-definition pairs which were assigned rather high probabilities, for example: P(**b**etween their

[2]When the amount of data for training and testing is limited, the n-fold method is used. The n-fold method divides the data into n equal parts, it uses n-1 for training and the remaining 1 for testing. This process is repeated n times, so that each 1/n-th part of the dataset is tested once and each instance in the data is used for training at least n-1 times.

body **m**ass **i**ndex (BMI))= 0.089. Those probabilities were utilized for the machine learning experiment which we describe in section 4. An example of a feature vector that describe the acronym-definition pair "**H**emocult **II** (H-II)" is: 1, 1, 0, 4, 11, 0, 7, 3, 3, 2, 0.803, +

Feature set

The original feature set described in Dannélls (2006a) consists of ten syntactic features, each of which describes an acronym-definition pair and its appearance in the context from which it was extracted. In addition to these features we include a feature which is intended to capture the local actions sequence that must be performed on the definition string in order to generate its acronym. This feature was obtained using a Maximum Entropy Markov Model (MEMM) approach (Tsuruoka et al., 2005), the model is introduced in the beginning of this section.

In total, eleven numeric features were used in this experiment, these features are as follows: (1) the acronym or the definition is between parentheses (0-false, 1-true); (2) the definition appears before the acronym (0-false, 1-true); (3) the distance in words between the acronym and the definition; (4) the number of characters in the acronym; (5) the number of characters in the definition; (6) the number of lower case letters in the acronym; (7) the number of lower case letters in the definition; (8) the number of upper case letters in the acronym; (9) the number of upper case letters in the definition; (10) the number of words in the definition; (11) the probability of the acronym assigned by the MEMM.

4 Experiment and results

We report the results for two experiments in which four classifiers were trained to predict the class of 134 instances, these are viewed in table 1. The middle column shows the results which were reported in Dannélls (2006a) and which are used as our baseline. These results are comparable since the experiment was performed with the same data set and feature set as in the experiment presented in this paper. The only difference is an additional feature, that is the eleven:th feature which contains the probability which was assigned by the Markov model as described in section 3. As the results show the new feature led to

Classifier	Accuracy (%) 10 features	Accuracy (%) 11 features
IB1	93.7	93.1
IGTREE	96.3	96.4
TRIBL	94.1	94.0
TRIBL2	94.5	94.9

Table 1: Memory-Based algorithm results

rather good results. In comparison to our baseline, the highest improvement achieved was by the TRIBL2 classifier which increased with 0.4 %, yielding 94.9 % accuracy. The reason

why the IB1 accuracy decreased can be explained by the fact that the IG weighted IB1 algorithm (IB1-IG) is relatively costly. For each test case, all feature values are compared to the corresponding feature values of all the training cases. The IGTREE algorithm, on the other hand, restructures memory in a compressed decision tree structure (van der Sloot, 2005).

5 Conclusion

In this work we conducted an experiment to test the assumption that the information sources of the training experience from which a learning system learns have a major effect on the learning results. Previous work with acronyms have proven the usefulness in approaching the acronym-definition recognition problem by utlizing different learning methods and various feature combinations. We repeated a machine learning experiment on a set of 671 acronym-definition pairs, the feature vectors for this dataset were recomputed with an additional feature which includes statistical information about each acronym-definition pair. Though we only increased the feature set with one attribute, this attribute is valuable as it contains comprehensive information about the acronym-definition pair. We examined the results of the learning models and compared them with previous result.

The results show learning becomes more accurate when the training examples contain representative attributes set, though there was an improvement of only 0.4% and it is hard to judge whether this has any significant for the effect on the training experiences and the choice of relevant information sources for the learning algorithm. The probability values for the negative and positive instances are rather confusing i.e., positive examples were assigned low probabilities, negative examples were assigned high probabilities. A larger data is required to train the MEMM-based tagger in order to obtain reliable results. Moreover, the training experience used in this experiment consists of 537 instances. This distribution of examples over which the final system performance must be measured may not follow a distribution similar to that of the future test examples. For more reliable results it is necessary to train the models presented in this paper on a larger distribution of examples. We believe the results can further improve by increasing the feature set and jointly optimize different feature selection and algorithm parameter (Daelemans et al., 2003).

Acknowledgement

We would like to acknowledge Yoshimasa Tsuruoka for providing us the trained model and the data.

References

Jeffrey T. Chang, Hinrich Schütze, and Russ B. Altman. Creating an online dictionary of abbreviations from medline. *Journal of American Medical Informatics Association (JAMIA)*, 9(6):612–620, 2002. http://abbreviation.stanford.edu/.

Walter Daelemans. Memory-based language processing. introduction to the special issue. *Journal of Experimental and Theoretical AI*, 3(11):287–467, 1999.

Walter Daelemans, Véronique Hoste, Fien De Meulder, and Bart Naudts. Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. In *In Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*, Cavtat-Dubrovnik, Croatia, 2003.

Dana Dannélls. Automatic acronym recognition. In *Proceedings of the 11th conference on European chapter of the Association for Computational Linguistics*, pages 167–170, 2006a. Trento, Italy.

Dana Dannélls. Acronym recognition: Recognizing acronyms in swedish texts. Master's thesis, Department of Linguistics, Göteborg University, Sweden, June 2006b.

Dimitrios Kokkinakis. Collection, encoding and linguistic processing of a swedish medical corpus: The medlex experience. In *Proceedings of the 5th Language Resources and Evaluation Conference*, 2006. Genoa, Italy.

Tom M. Mitchell. *Machine Learning*. MIT Press and The McGraw-Hill Companies,Inc, 1997. Singapore.

David Nadeau and Peter Turney. A supervised learning approach to acronym identification. In *Proceedings 18th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 319–329, 2005. Victoria, BC, Canada.

Yoshimasa Tsuruoka, Sophia Ananiadou, and Jun'ichi Tsujii. A machine learning approach to acronym generation. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*, Mining Biological Semantics, pages 25–31, 2005. Japan.

Ko van der Sloot. Timbl: Tilburg memory-based learner. Technical Report version 5.1, Computational Linguistics Tilburg University, January 2005. API Reference Guide.

Manuel Zahariev. *A Acronym*. PhD thesis, Simon Fraser University, 2004.