Exploiting Maching Learning for Automatic Semantic Feature Assignment

Karel Bílek, Natalia Klyueva, Vladislav Kuboň
Charles University in Prague
Czech Republic

Abstract

In this paper we experiment with supervised machine learning techniques for the task of assigning semantic categories to nouns in Czech. The experiments work with 16 semantic categories based on available manually annotated data. The paper compares two possible approaches - one based on the contextual information, the other based upon morphological properties - we are trying to automatically extract final segments of lemmas which might carry semantic information. The central problem of this research is finding the features for machine learning that produce better results for relatively small training data size.

Introduction

Lexicons enhanced with semantic information are frequently used in various NLP applications, such as machine translation, question-answering or sentiment analysis. Probably the most well known resource of such kind is WordNet (Fellbaum 1998), the lexicon of words interlinked by semantic relations and organized hierarchically into semantic classes. It nowadays exists for many languages including Czech. Although it is a large scale resource providing complex semantic information, it's applicability is often limited by the fact that it was created by means of a translation of the English WordNet and that it uses a system of categories which may not fit a particular application.

Many additional tools for automatic semantic annotation have been created so far, as, e.g., for semantic relation assignment (Peirsman 2011) or multipurpose semantic memory (Baroni and Lenci 2009) etc.

In this paper we describe experiments with automatic assignment of semantic features (categories) to Czech nouns exploiting an existing resource (a small hand-annotated lexicon created originally for a machine translation system). The assignment is performed by means of logistic regression models. The model is trained in a supervised manner, using basically two kinds of features for machine learning – morphological and syntactic (context behaviour) properties and their combination. The experiments suggest an answer to a question which type of features is more useful for the given purpose.

Note on Notation

Since the word "feature" is frequently used both in linguistic context as a term for semantic category and in machine learning context as a term for describing any observable property of a learning example, for better clarity, we use the term "category" for the notion of semantic feature, while we use the word "feature" purely for machine learning feature in the rest of this paper.

Motivation

Semantic Categories

Semantic categories are generally viewed as components of meaning that express one definite sense of a word, they are generally associated with the contrastive context, so they occur either with plus or minus sign, ex. [+human], [-time] etc. There is no generally accepted set of semantic features, as researches usually use their own classification depending on their goals. Assigning semantic features to concrete words is influenced by a general issue of finding the proper level of granularity - if the number of semantic features is relatively low, they are not rich enough to capture more subtle differences in the woird meaning; if too many of them are used, it is more difficult to assign them correctly for each particular word. This is probably one of the reasons why the number of distinct semantic features varies in various experiments from only a few (like the division of nouns into 'animated' or 'non-animated') to the very fine-grained meaning classification (as, e.g., in WordNet).

Semantic categories can also be used for some minor research problems, as, for example resolving nominative-accusative ambiguity in Slavic languages. In (Justeson and Katz 1995) the authors show the way to disambiguate adjectives with the help of semantic categories of nouns they modify.

For example, it may help to discriminate two senses of the adjective *short* : applied to those nouns with the sem. category [+human] versus in combination with a [+interval] noun. The phrase *a short girl* is translated into Czech as *malá holka*, whereas *a short day* is *krátký den*.

It can also help to disambiguate different senses of verbs:
(1)*The dog runs after the owner.* - [+human] category of a subject
(2) *The program runs on Linux.* -[+computer] category of a

subject
If translated for example from English or Czech into Russian, the disambiguation is needed: while the verb in (1) can be translated straightforwardly as *bežať* - 'to run' into Russian, the metaphorical meaning in (2) must be expressed by another verb – *rabotať* - 'to work'.

Classification

The task of semantic category assignment can be naturally represented as a task of classification, using supervised machine learning algorithms.

Generally, in supervised machine learning, we have a set of training data with defined categories and we want to build an algorithm, which will generalize from this training data and return a category for any yet unseen data.

More specifically, in most of the algorithms, the training data can be broken into *examples*, and each of the examples can be described as a set of *features* and a given *category*. The machine learning algorithm is usually a model with variable parameters, which are then learned from the training data.

With the data we have at our disposal, we can take the semantically annotated words from the lexicon as training examples and their linguistic properties as features in machine learning.

Sources

Semantic Categories

Reliable training data constitute the necessary condition for the success of any supervised machine learning algorithm. Instead of creating a new set of semantic categories and subsequently undertaking a long and costly process of manual annotation, we have decided to re-use existing high quality manually annotated data. Such data exist in the form of a bilingual dictionary of a machine translation system RUSLAN, a rule based MT system translating from Czech to Russian.

The history of RUSLAN (Oliva 1989) goes back to the second half of the 80's when there was a need for an automatic translation of operating systems manuals. However, due to the political changes after 1989, there was no need for such MT between Czech and Russian anymore. Since then, the resources used in the project served mainly as a source of data for other projects, for example, in (Bojar, Homola, and Kuboň, V. 2005) authors tried to re-use the module of syntactic analysis of Czech for the Czech-English Machine Translation, the paper (Klyueva and Kuboň, V. 2010) describes the extraction of morphosyntactic information from the bilingual dictionary of RUSLAN for Czech and Russian valency dictionaries.

The bilingual dictionary of RUSLAN constitutes a rich source of various kinds of data for both Czech and Russian. Apart from the target language equivalents and their morphology, the dictionary provides morphological, syntactic and semantic information for the source language (Czech), namely declension patterns, valency frames, semantic features etc. In the current experiment we ignore the Russian (target language) part of the dictionary and we use (a relevant part of) the Czech side, namely the semantic features assigned to all nouns in the dictionary.

Shortcut	Count	Category
A	941	abstract
C	835	activity
R	728	result
K	712	concrete
V	205	property
H	165	animated
Z	101	machinery
M	64	measure
P	56	program
N	44	instrument
F	41	function
D	32	action

Table 1: Semantic categories in training data

Although the high quality of human annotation of the words in the dictionary is a valuable asset, the dictionary of Ruslan also has one drawback - a relatively limited domain. The project aimed at the translation of manuals to operating systems of mainframes. It thus contains a relatively smaller number of domain-independent words, the domain related expressions prevail. On top of that, even the computer terminology has changed during the past 25 years, so some of the words contained in the dictionary are slightly outdated.

Reusing Data from RUSLAN Dictionary The dictionary of RUSLAN contains about 8,000 entries, 2,783 of which are nouns with assigned semantic categories.

Following is an example of an entry:

```
LE2KAR3==MZ(@(*H),!,MA0111,VRAC2).
```

- `LE2KAR3` represents the Czech lemma *lékař*; the diacritics is encoded in a rather primitive way corresponding to the time when it was created and when encoding of national characters still constituted a challenge.
- `MZ` represents a declension pattern (and thus also determines the part of speech information because this particular declension pattern is used for masculine animate nouns in Czech).
- `@(*H)` represents the semantic category 'animated'.
- `MA0111,VRAC2` represents the declension class of the Russian equivalent and the equivalent itself, encoded into basic ASCII

From this format we have extracted the Czech side of the dictionary together with the semantic categories. When we took the categories without "sanity checking" and filtering out the possible mistakes, we ended up with 2,783 words and 29 categories; however, some of these categories appear only with one or two words, therefore they are not relevant for our purpose. When we filter out those categories, that don't appear in at least 30 words, we ended up with the features described in Table 1.

The total number of categories assigned to words is bigger than the number of words because some words have more than one semantic category. In average, each words is assigned 1.4 categories.

Monolingual Data Corpus

Because one of our methods was supposed to exploit an immediate context, it was necessary to use additional annotated corpora. As the highest quality annotation of Czech is provided in the Prague Dependency Treebank (PDT),[1], it was a natural first choice.

PDT PDT represents a collection of Czech texts annotated on three levels - morphological, analytical(surface syntactic) and tectogrammatical(deep syntactic). It contains 115,844 sentences from newspapers and journals.

However, we have found out that even the size of PDT does not provide a sufficient coverage of the words from the RUSLAN domain. The 115,844 sentences of PDT contain 1,957,247 tokens, but out of the 2,783 words from RUSLAN, 813 don't appear at all in the entire corpus, 162 appear only once and 1,408 words appear less then 10 times.

These numbers clearly indicate that the contextual information based on manually annotated data from RUSLAN is too sparse for machine learning. In would actually mean that many feature vectors would be empty which would cause many examples to be misaligned completely.

For that reason, we decided to use a bigger corpus with only morphological annotation.

WebColl WebColl(Spoustova, Spousta, and Pecina 2010) is a corpus of texts in Czech crawled from the Czech web, cleaned and annotated with a POS tagger and lemmatized. WebColl consists of 7,148,630 sentences, which together have 114,867,064 tokens.

Although this corpus is about 100 times bigger than PDT, its data cover our lexicon only slightly more. Out of the 2,783 words, 412 don't appear at all, 40 appear exactly once and 611 words appear less than 10 times. In other words – increasing the training data size approximately 100 - times results in the removal of only about a half of the unseen words.

The manual revision of the unseen words revealed that most of those words are very domain specific (words such as "*rebasing*", "*subroutine*", "*self-relocability*" and so on) and that they probably won't appear frequently enough no matter how big corpus we take. With regard to the rest, some of those words were genuine mistakes and some of them were affected by slightly different lemmatization used in RUSLAN and WebColl (incompatible lemmas).

Machine Learning Features

The success of machine learning algorithms to great extent depends on the choice of proper features. In our experiments we have tried to exploit two types of features - context ones and morphological ones. In other words, we exploited syntax and morphology in order to learn semantics of a word.

[1] http://ufal.mff.cuni.cz/pdt2.0/

Context

As J. R. Firth stated "Words shall be known by the company they keep", therefore our first idea (and the main reason for using a large mono-lingual corpus) was to look at the context in which the words appear and to try to convert it into machine learning features.

The context can in principle be exploited in a number of different ways. For example, in (Baroni and Lenci 2009), the authors proposed a scheme to retrieve various semantic properties of words from the context. In (Biemann and Osswald 2005) the semantic features of nouns were learned from a context, but only adjectives were taken into account. We have decided to use all parts of speech as a context in our experiment. This decision was motivated by the endeavor to use as much information from the context as possible.

We have taken into account the context of two words to the left and two words to the right. First of all, it was necessary to determine the context of all "known" words from the RUSLAN dictionary, i.e. the words whose semantic categories have been assigned manually and thus can be considered reliable and correct.

For every word from the dictionary, we have looked at all words in its context. If n types were found, then it actually meant that we have obtained n separate features for that particular word, where the value of each feature represents the number of times when the feature word appeared in the context.

The machine learning model was then trained on these features. To assign a category to an unseen word, we had to go through the entire corpus, count the features for the unseen word as well using the same algorithm as for the features of the known words. However, this time we were not collecting all words appearing in the left or right context of the unseen word, but only those appearing among the features collected for known words (in other words, we counted all the words in the context and then performed an intersection of the counted words and the words from the features). Again, the number obtained for each feature represented the value of that feature for the unseen word and became a part of the feature vector of the machine learning model.

This naive approach had several drawbacks. Most importantly, our number of features exploded, while the values (number of appearances) themselves were very unevenly distributed.

For this reason we made several adjustments:

- we nominated a context word as a feature only when the given context word was seen in at least some fixed number of training examples min. The motivation for this decision was very simple - it a given word appears in the context only few times, it does not tell much about the context (a "training example" here means a word from the dictionary)
- we normalized the numbers, so that the features were all approximately of the same size. We originally wanted to use percents as values - meaning, we wouldn't have the number of appearances as a feature, but the percentage of how often is a given word in the context of the training example. However, this resulted in very small numbers,

since actually, in most of the training examples, even the top context features are in the order of fractions of 1 percent; so, at the end we have decided to use percentage multiplied by ten and converted to integers. We ignore the values when the integer value is smaller than 1 (that means, if it is the context in less than 1/1000 of cases).

If we set min as 40, the number of features is 1,502.

Morphology

The second approach we have decided to try was based upon morpho-semantic properties of nouns. The main idea of this experiment was based upon the investigation carried out by Z.Kirschner in his system MOSAIC (Kirschner 1983). He exploited the fact that many suffixes in natural languages determine the semantic nature of words. For example, in English, the suffixes *-or* or *-er* usually appear with words having a semantic role of an actor of some activity, *-tion* is an activity, *-ity* or *-ness* marks a property. In Czech, *-ič*, *-ač*, *-čka*, *-ér*, *-or*, *-dlo*, *-metr*, *-graf*, *-fon*, *-skop* are tools or machines, *-ace*, *-kce*, *-áž*, *-ní*, *-za* processes or activities, *-ost*, *-ita*, *-nce* properties; and adjectives ending with *-aný*, *-ený* are results of processes while *-ací*, *-ecí* marks a purpose.

There were two major reasons why we could not apply the method from MOSAIC directly. First, the semantic categories determined by Czech suffixes do not directly correspond to the set of semantic categories we are using in our experiments; and, second, the number of suffixes seems to be too large ((Kirschner 1983) says that a full coverage of Czech technical texts would require about 2000 suffixes) and contain too many exceptions (for each domain it is necessary to create a dictionary of hand-picked exceptions, words, which have the particular suffix, but which do not belong to a semantic category usually marked by the suffix).

In order to avoid the long and costly process of manual selection of relevant suffixes, we have decided to rely on data and to try adding suffixes as a machine learning feature. More precisely, we took the last n letters from a word (for $n = 4,3,2$ and even 1) and we then created a new feature for every such suffix.

It actually resulted in the feature vector for any training example consisting mostly of zeroes (it has 1 only for one ending of length 4, one ending of length 3, etc., all the other features being equal to 0). On the other hand, the advantage of such brute-force approach is obvious - it is much easier to find such feature for a new word, because it is fully determined by the word itself and it is not necessary to use any other source of information.

Also the morphological features tend to explode quickly, therefore it was necessary to apply the same measures to stop feature explosion - we used only those suffixes that appeared at least min-times in the dictionary.

With min set to 5, the number of features reached a reasonable value of 433.

Suffixes and Semantic Categories

The identification of relevant suffixes is not sufficient, as a second step it is also necessary to link the suffixes with relevant semantic categories. Below are some examples of endings that can indicate the semantic features of words according to our experiments.

A abstract	*ivost, ekce, ování, íra, ita, nictví*
H animated	*átor, ovník, atel, atelka, stník, or, ík*
C activity	*ení, ování, ání, ace*
D action	*ení, ání, ace*
R result	*ita, ení, utí, ání, akce, utí*
K concrete	*ka, íč, énko, ora*
M measure/unit	*etnost, ita, ví, o, kost, etí*
V property	*ita, nost, ce*

Table 2: Sem.categories and the endings they take most frequently

Machine Learning Appropaches

Logistic Regression

The most suitable tool for our experiments seems to be the logistic regression. It can predict to which category a particular unseen word belongs. If we have several binary classifiers, it is recommended to use *one-vs-all* models, with the category with the biggest chance "winning". However, this model is only applicable to the case where we have ***single category*** with every training case. In our case, though, we have multiple categories and their assignment is therefore not so straightforward.

In order to solve this problem, we have employed a simple solution, where every category has its own classifier trained separately, returning 1 or 0 and thus indicating whether a particular input word belongs to this category or not. The item is then indicated as belonging to all the categories where the classifier returned 1.

For every experiment, we put aside the same (at the beginning randomly selected) set for testing purposes. On the training set, we put aside a heldout set, and we train the parameters λ and γ on the heldout set, simply by training the model with the given λ and γ, counting the F-score and getting the best λ and γ for the given classifier.

These parameters vary from classifier to classifier; for a given feature set, we train 14 classifiers (one for every semantic category), together with training λ and γ parameters, and then we test those categories on the test set.

Machine Learning Evaluation

Because we use multi-labeled classification, we cannot use only precision and recall. Instead, we use so-called micro-average and macro-average of both precision and recall, and then make their mean average for micro-averaged and macro-averaged F-score. We use μ as a symbol for micro and M as a symbol for macro.

Micro- and macro-averaged precision and recall is defined as [2]

[2]From Data Mining wiki `http://datamin.ubbcluj.ro/wiki/index.php/Evaluation_methods_in_text_categorization`, Babes-Bolyai University, Romania

$$P_\mu = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i}; \quad R_\mu = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i}$$

$$P_{\mathrm{M}} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FP_i}; \quad R_{\mathrm{M}} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FN_i}$$

where TP_i is true positive given a category i and C is the set of categories. Basically, micro-averaging gives equal weights to every document, while macro-averaging gives equal weights to every category.

Results

First of all, let us present the results achieved for the baseline experiment and for both experiments using the above described sets of features. All results are rounded to two significant digits.

Logistic Regression

Random Baseline For comparison, we began with two "baseline" algorithms - first it was a random classifier, that didn't look at the features at all; the second was logistic regression model, but trained on randomly generated 200 columns of random "features".

	P_μ	R_μ	F_μ	P_M	R_M	F_M
Rand. class.	0.11	0.51	0.19	0.11	0.57	0.18
Rand. feats.	0.30	0.36	0.32	0.10	0.14	0.12

Context In order to find out whether the context based method provides enough information for semantic classification, we have not tested all possibilities (various length of the context, different threshold value min etc.). In the first experiment we have taken only 1 word on the left and 1 word on the right as a context feature, and we have set min to 40. This gave us 980 features. With those, the results are as follows:

	P_μ	R_μ	F_μ	P_M	R_M	F_M
Dist. 1	0.27	0.57	0.36	0.23	0.48	0.30

The second experiment extended the context to words with distance 2 both to the left and to the right (with min still 40). This increased the number of features only slightly, to 1,501. The results were unfortunately not much better, thus indicating that further extension of the context is most probably irrelevant:

	P_μ	R_μ	F_μ	P_M	R_M	F_M
Dist. 2	0.28	0.62	0.39	0.23	0.49	0.32

	P_μ	R_μ	F_μ	P_M	R_M	F_M
Morphology	0.57	0.66	0.61	0.46	0.41	0.43

Morphological Features As the next step, we have tested the logistic regression with morphological features obtained in the way described above. The results are quite encouraging compared to the experiments with the context, they constitute a substantial improvement:

If we break the results into individual classifiers, we can see that for several categories it is quite successful with relatively high F-score, thus indicating that certain suffixes related to some categories are really quite productive and that the assumption that they carry the meaning of the word is probably right. On the other hand, 0 F-score with several categories represents a complete failure of the mechanism. It happened due to a very small number of examples for those categories in the *testing* data, and due to the fact that they all got misaligned. This result might also indicate that the set of semantic categories was itself only partially well-chosen. Some categories are too seldom in order to be taken into account, certain redefinition might be useful for future experiments. The exact results are presented below:

abstract	0.57
activity	0.75
result	0.66
concrete	0.58
property	0.78
animated	0.57
machinery	0.11
measure	0.18
program	0
instrument	0
function	0
action	0.40

Combination of Approaches

The encouraging results of the morphological classifier led to the third experiment. The main question of this experiment was whether the context can add some information to the suffixes. We have tested two possible ways how to combine context and morphology. The first one was a bit more sophisticated, we tried to include the morphology of the context. That means - we tried to add features as "the sum of the combinations of last 3 letters of the words with the distance max. 2 to the right", and so on.

We originally wanted to let the min value 40 to keep the experiments consistent, but the feature space exploded quickly. So, as a way of a very primitive feature cutting, we increased the min value for those features where we take more letters into account.

The endings and beginnings of context words with the distances 1 gave us these results: (We took min as 40 for the lengths 1, 80 for the length 2 and 140 for the length 3. Those numbers were quite arbitrarily chosen to keep the feature space relatively small in order to prevent the algorithm becoming too slow and the feature space too big.)

With the added 3rd letter, the F-scores are almost the same

letters	feats	P_μ	R_μ	F_μ	P_M	R_M	F_M
1	138	0.22	0.47	0.30	0.20	0.43	0.29
1, 2	901	0.25	0.54	0.35	0.22	0.48	0.30
1, 2, 3	1659	0.26	0.54	0.35	0.23	0.48	0.31

as for less complicated combinations while the feature space has grown substantially, making our algorithm slower. The worst aspect is, however, that the results are actually even worse than the results for the context alone.

We can only wonder whether the reason why the context is not working as expected, is that the feature space becomes unrealistically big and the so-called Curse of Dimensionality starts to take place.[3] Basically, we get a huge matrix with more features than training examples, but the matrix is very sparse.

However, even after a substantial reduction of the space in subsequent experiments, we were never able to get better results from this ' 'combined" approach than from pure context. This means that all we can get out of context is best done by looking at the context alone, not at its endings or beginnings.

Combination - Putting Features Together The second type of combination is actually more primitive than the first one. It simply takes both types of features and puts them together. However, neither this simple combination provided better results than morphology alone, further underlining the fact that the morphology of the words seems to be more useful than context.

	P_μ	R_μ	F_μ	P_M	R_M	F_M
Combined	0.54	0.70	0.61	0.33	0.42	0.37

Conclusion

In this paper we have described the experiments in applying machine learning algorithms to guessing the semantic category of nouns based on the following features:
1. A (limited) context surrounding a word
2. Morphological characteristics of a word, namely its suffix

We have shown that context properties of words provide less information than suffixes, both alone and in combination. Even a very simple method of guessing semantic features on the basis of the surface form of a word brought more favorable results in terms of precision. It was also an interesting observation, that adding context features to the morphological ones in machine learning negatively influenced the results.

There are several possible directions for further research. One lies in the field of optimization of features for Machine Learning algorithm, the second one might explore adapting WordNet as training and testing data for our task. The third one points towards designing non-trivial context features taking into account part-of-speech tags and other information. Last but not least is the effort to obtain a bigger number of hand-annotated data of a more general nature than the rather domain specific data we had at our disposal in these experiments.

[3]This is sometimes known as "Hughes effect", see (Hughes 1968)

Acknowledgments

The research was supported by the grants GACR P406/2010/0875 and GAUK 639012.

References

Baroni, M., and Lenci, A. 2009. One distributional memory, many semantic spaces. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, GEMS '09, 1–8. Stroudsburg, PA, USA: Association for Computational Linguistics.

Biemann, C., and Osswald, R. 2005. Automatic extension of feature-based semantic lexicons via contextual attributes. In *Proceedings of the 29th Annual Conference of the Gesellschaft fA 1/4 r Klassifikation e.V., University of Magdeburg, March 9-11, 2005*, 326–333. Springer.

Bojar, O.; Homola, P.; and Kuboň, V. 2005. An MT System Recycled. In *Proceedings of MT Summit X*, 380–387.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Hughes, G. 1968. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory 14 (1)* 55–63.

Justeson, J. S., and Katz, S. M. 1995. Principled disambiguation: Discriminating adjective senses with modified nouns. *Computational Linguistics* 21(1):1–27.

Kirschner, Z. 1983. *MOSAIC, a Method of Automatic Extraction of Significant Terms from Texts*. Explizite Beschreibung der Sprache und automatische Textbearbeitung. Matematicko-fyzikální fakulta Univerzity Karlovy.

Klyueva, N., and Kuboň, V. 2010. Verbal valency in the mt between related languages. In *Proceedings of Verb 2010, Interdisciplinary Workshop on Verbs, The Identification and Representation of Verb Features*.

Oliva, K. 1989. *A Parser for Czech Implemented in Systems Q*. Explizite Beschreibung der Sprache und automatische Textbearbeitung. Matematicko-fyzikální fakulta UK.

Peirsman, Y.and Padó, S. 2011. Semantic relations in bilingual vector spaces. *ACM Transactions in Speech and Language Processing* 3:1–3:21.

Spoustova, J.; Spousta, M.; and Pecina, P. 2010. Building a web corpus of czech. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).