Introduction to Machine Learning

Term project specification

PFL054 2011/12

Semantic Pattern Classification

The aim of the term project is to solve "the best you can" classification task. Your classifiers will work with English verbs and their one-sentence contexts and should recognize semantic patterns of verb usages. Each occurrence of the selected verbs should be classified into a given set of semantic patterns according to the specific verb's context.

1. What are semantic patterns?

Traditional lexicographers usually assume that various uses of polysemous words can be sorted into discrete *senses*. When building a dictionary entry for a given word, the lexicographer sorts a number of its occurrences into discrete senses present (or emerging) in his/her mental lexicon, which is supposed to be shared by all speakers of the same language. The assumed common mental representation of a word's meaning should make it easy for other humans to assign random occurrences of the word to one of the pre-defined senses (Fellbaum et al., 1997). However, this approach to *lexical disambiguation* has turned out to be very difficult, especially because it has never been clear how to define the ''right'' list of senses for a given word. The very notion of ''word

sense'' is slippery and controversial (see e.g. Kilgarriff (1997)). For humans, it is hard to make agreement about the ''right'' senses, and for computers it is hard to perform the classic task of *word sense disambiguation* (WSD), which means to automatically assign the ''correct'' sense to a word occurring in a given particular context. On finer-grained sense distinctions, top accuracies from 59% to 69% have been reported in recent evaluation exercises, where the baseline accuracy of the simplest possible algorithm of always choosing the most frequent sense was about 54%. Even interannotator agreement between humans is usually quite low. A brief overview of the WSD field is available at http://en.wikipedia.org/wiki/Word-sense_disambiguation.

Semantic pattern recognition (SPR) is a novel, alternative approach to lexical disambiguation, which is different from the traditional word-sense assignment tasks. The SPR approach does not assume that words have fixed senses but that regular patterns of their usage can be identified in a corpus, and that the patterns activate different conversational *implicatures* from their *meaning potentials* (Hanks and Pustejovsky, 2005).

In this project we will focus on English verbs and will use the *Pattern Dictionary of English Verbs* (PDEV) (Hanks and Pustejovsky, 2005). PDEV is a database of manually extracted patterns of frequent and normal verb uses. The patterns are, roughly speaking, intuitively similar uses of a verb that express—in a syntactically similar form—similar events in which similar participants (e.g. humans, artifacts, institutions, or other events) are involved. Two patterns can be semantically so tightly related that they could appear together under one sense in a traditional dictionary. The patterns are not senses but syntactico-semantically characterized prototypes (see the example verb *submit* in Table 1). A few examples can be found at https://wiki.ufal.ms.mff.cuni.cz/external:spr.

No.	Pattern / Implicature
1	**[[Human 1 | Institution 1] ^ [Human 1 | Institution 1 = Competitor]] submit [[Plan | Document | Speech Act | Proposition | {complaint | demand | request | claim | application | proposal | report | resignation | information | plea | petition | memorandum | budget | amendment | programme | ...}] ^ [Artifact | Artwork | Service | Activity | {design | tender | bid | entry | dance | ...}]] (({to} Human 2 | Institution 2 = authority)^({to} Human 2 | Institution 2 = referee)) ({for} {approval | discussion | arbitration | inspection | designation | assessment | funding | taxation | ...})** [[Human 1 | Institution 1]] presents [[Plan | Document]] to [[Human 2 | Institution 2]] for {approval | discussion | arbitration | inspection | designation | assessment | taxation | ...}
2	**[Human | Institution] submit [THAT-CL|QUOTE]** [[Human | Institution]] respectfully expresses {that [CLAUSE]} and invites listeners or readers to accept that {that [CLAUSE]} is true}
4	**[Human 1 | Institution 1] submit (Self) ({to} Human 2 | Institution 2)** [[Human 1 | Institution 1]] acknowledges the superior force of [[Human 2 | Institution 2]] and puts [[Self]] in the power of [[Human 2 | Institution 2]]
5	**[Human 1] submit (Self) [[{to} Eventuality = Unpleasant] ^ [{to} Rule]]** [[Human 1]] accepts [[Rule |Eventuality = Unpleasant]] without complaining
6	**[passive]** **[Human| Institution] submit [Anything] [{to} Eventuality]** [[Human 1|Institution 1]] exposes [[Anything]] to [[Eventuality]]

Table 1: Example of patterns defined for the verb *submit*.

Motivation

Lexical disambiguation is a traditional task of corpus linguistics and natural language processing. The goal is to recognize different meanings of polysemous words in particular contexts. Lexical disambiguation is important and can be used in many subfields of applied computational linguistics, e.g. in computational lexicography, textual entailment, discourse analysis, question answering, machine translation, or information retrieval.

2. Your task in detail

In this project, you will deal only with the following 6 verbs: *ally, arrive, cry, halt, plough, submit*. Their patterns are available in the directory data/patterns. For each verb you will implement a supervised classifier.

As your training (development) data you will have a set of 250 manually annotated sentences for each verb, which are available in the directory data/development-instances. Each sentence contains one of the selected verbs, and is considered as a data instance to be classified. First, you should make feature vectors to describe all data instances. Then you will choose a suitable method of machine learning, design and implement a classifier, and tune its parameters. Finally, when you submit your code, it will be evaluated on ''unseen'' test sets (50 test sentences for each verb).

Two tasks to be completed

You will have to work on and complete two tasks, the basic one and the advanced one:

- **Basic task (A)**

  In this task you will process all 6 verbs and will use the default feature set (see below). You have right to reduce the defined default feature set (and use some its subset instead), but you must not extend it. You should experiment and then choose one model (i.e. one machine-learning method) that will be then used for all 6 verbs. However, the best parameters of the model can and will be learned for each verb separately. So you will get 6 different classifiers, all based on the same machine-learning method.

  Only this task will be reported in your short report.

- **Advanced task (B)**

  In the advanced task (B), setting a suitable set of features used for classification will be a part of the project. After you have developed your

best model using the default feature set, you will choose 3 of the 6 verbs (it will be *your* choice), and then you will try to develop a better feature set for each of the 3 selected verbs to improve the classifier performance. Of course, you can develop a model totally different from the one used in the task (A). However, your new model should be still the same for all 3 verbs that you chosen (although feature sets can be different for each of the 3 verbs). You should compare the results obtained in the task (A) with the new results. Everything about both tasks (A) and (B) should be exactly documented in your final report.

Competition

Your work will be viewed as a competition. All you students will get the same annotated data. Then you will choose (some of) standard machine learning methods and make your experiments. You will tune the parameters of your classifiers and analyse and compare their performance. Finally you will choose the classifier that you consider to be the best for the given task.

When you finish all your work and your program codes, you will submit your final solution in the form of a detailed report. The report should contain both the description of the methods used (including the description of parameters tuning) and the analysis of the results. Conclusion of your final report will include your choice of the best model you have developed. You should compare at least three machine learning methods and choose the best one regarding the quality of their output measured on your development test data.

Your best classifiers will be evaluated on our test sets (that will be hidden from you until you submit your final classifiers). The student with best results on our test data will be the winner (one winner of the task (A), and one (possibly different) winner of the task (B)).

Evaluation

The task (A) will be evaluated using the weighted average of the accuracy of your 6 classifiers. The weights correspond to the relative frequencies of the 6 verbs in the BNC50 corpus (cf. Cinková et al. (2010)). The coverage of all verb occurrences in the BNC50 by the selected verbs is 0.0083% (ally), 0.1307% (arrive), 0.0257% (cry), 0.0183% (halt), 0.0076% (plough), 0.0483% (submit).

You should optimize the average accuracy calculated as

$$\sum_v p_v * A_v \,/\, 0.2389\%,$$

where p_v are the verbs' relative frequencies, A_v are the accuracies of your 6 classifiers, and 0.2389% is the sum of the relative frequencies.

The task (B) will be evaluated using the proportion of the number of people who you have beated and the number of people who have beated you. When you submit your classifiers for 3 verbs, for each of those verbs we will count the people who submitted their classifiers for the same verb and had a better accuracy, and the same for submitted classifiers with a worse accuracy. Then the counts will be summed up and the proportion of the two sums will be your score. The best score wins.

3 Data description

Primary data

All annotated sentences have been (randomly) selected from the BNC corpus. You will get the data only for internal purposes of our course. Please, use the provided data only for your study or academic purposes. You are not allowed to distribute it.

The manually annotated data sets are stored in 6 text files (one file with 250 sentences per verb) in the directory data/development-instances. Each instance consists of 6 lines/items:

- *sentence ID* – you do not need it;
- *pattern tag* – which is, in fact, the manually annotated *class label*;
- *tokenized sentence* with marked *target verb* – tokens are separated by spaces;
- *morphologically analysed sentence* – tokens are separated by tabs; each token includes 1) original word form, 2) its lemma, and 3) its morphological tag; both lemmas and morphological tags have been determined automatically (thus, some errors can occur); the description of morphological tags is available in the Appendix A;
- list of *syntactic dependencies* – obtained automatically using the Stanford dependency parser (the format ''collapsed dependencies with propagation of conjunct dependencies); an example sentence is given in the Appendix B; a detailed description of the Stanford dependency types is provided in the attached manual (the file stanford-dep-manual.pdf);
- output of a Named Entity Recognizer (probably you will not use it, even though you can).

Pattern tags

Note that some of the annotated sentences are marked with only a *pattern number* (they show *normal patterns*, i.e. represent ''perfect matches'' with pattern definitions), while some other were assigned a pattern number followed by a character (so called *exploitations*, i.e. deviations from the prototypical patterns of several different types: ''.a'' or ''.s'' or ''.c'' or ''.f''). Moreover, some of the sentences are marked with ''u'' (unclassifiables) or ''x'' (noise, not verbs to be tagged). For the very details you can see the attached *annotation manual* (in the file CPA_valiman.pdf).

IMPORTANT: In this project, ignore both a) the difference between normal patterns and exploitations, and b) the difference between ''u'' and ''x''. You will simply always assign either a pattern number or ''ux'' (standing for ''u or x'').

Development set and test set

You will get 1,500 manually classified instances (250 per verb). There are another 300 instances that make *test sets*, which you cannot see until you finish your "best classifiers" and submit your final report. Then your classifiers will be evaluated using the test sets.

Our *recommendation* is to split your data (for each verb) in two parts, a *development working set* and a *development test set*. When you develop your classifiers, you will use the development test sets both to evaluate your classifiers and to tune their parameters. Once you have finished your parameters tuning, you will choose the best model and use all the annotated training data (i.e. all 250 instances per verb you have got) to train your final, "best" classifiers. Those final classifiers will be submitted and then evaluated on the unseen test sets (the unseen 50 instances per verb).

4. Default feature set

Each data instance to be classified consists of the *target verb* (TV) and some context (*context words*). Therefore the values of the features that describe data instances will be based on the observed characteristics of both the TV and the context words.

You will use two kinds of features, the *morpho-syntactic features* and the *semantic features*. All features in the default set will be either binary (T/F values) or categorical (listed, discrete, non-numerical values).

I. Morpho-syntactic features

There are 83 morpho-syntactic features in the default feature set. 79 of them are binary, while the other 4 are categorical. Categorical features *can* be transformed into a set of binary ones.

- **1) Characteristics of the TV**

  TV itself wil be described by the following 10 binary features:

  - passive voice – presence of *auxpass*(TV, *)
  - modality1 – presence of *aux*(TV, *would* | *should*)
  - modality2 – presence of *aux*(TV, *can* | *could* | *may* | *must* | *ought* | *might*)
  - negation – presence of *neg*(TV, *)
  - tense
    - presence of the VBN tag assigned to the TV
    - presence of the VBD tag assigned to the TV
    - presence of the VBG tag assigned to the TV
    - presence of the VBP tag assigned to the TV
    - presence of the VB tag assigned to the TV
  - use in an infinite phrase (outside subject) – presence of *xcomp*(*, TV)

- **2) Characteristics of the words that immediately precede or follow the TV (simply by word order)**

  9 binary features will be established for each of the 6 closest context words: 1, 2, and 3 positions before and after the TV; so in total it will be 54 binary features; their values will depend on the presence of one of the listed morphological tags assigned to the 6 context words:

  - nominal-like (NN, NNS, NNP, NNPS, DT, PDT, PRP, PRP$, POS, CD)
  - adjective (JJ, JJR, JJS)
  - verbs (VB, VBD, VBG, VBN, VBP, VBZ)
  - modal (MD)
  - adverbial (RB, RBR, RBS, RP, IN)
  - ''to'' (TO)
  - wh-pronoun (WDT, WP, WP$)
  - wh-adverb (WRB)
  - to_be (lemma = ''be'')

- **3) Characteristics of the words that syntactically directly depend on the TV (according to the output of the Stanford dependency parser)**

  - **3A) Logical subjects**

    3 binary features:
    - *nsubj*(TV, *) - presence of a nominal subject
    - *csubj*(TV, *) - presence of a clausal subject

    Note that IF you find *xsubj*(TV, arg) (= a controlling subject) OR a*gent*(TV, arg) (a logical subject introduced by the preposition "by"), THEN you should take the arg as a subject:
    IF the arg is a noun or number or pronoun (= marked as NN* | CD | WDT | WP)
    THEN take it the same way as *nsubj*,
    ELSE take it the same way as *csubj*.

- plural_sb - presence of any subject in the plural form (see the morphological tag of the subject (if any is found) and test if it is NNS or NNPS)

- **3B) Objects**
  8 binary features:
  - *dobj*(TV, *) - presence of a direct object
  - *iobj*(TV, *) - presence of an indirect object
  - *nsubjpass*(TV, *) - presence of a passive nominal subject; (in fact, it is an object)
  - *csubjpass*(TV, *) - presence of a passive clausal subject; (in fact, it is an object)
  - *ccomp*(TV, *) - presence of a clausal complement (functions like an object of the verb)
  - *complm*(TV, *) - presence of a complementizer (typically the subordinating conjunction "that" or whether")
  - object - presence of any object (any of the above)
  - plural_obj - presence of any object in the plural form (see the morphological tag of the object (if any is found) and test if it is NNS or NNPS)

- **3C) Particles**
  If you find *prt*(TV, p), save the phrasal verb particle p as a categorical value. All possible values of this categorical feature will be the values found in the development working data + two special values: NONE and OTHER. (Beware of the fact that in the test data a new word can occur that you have not met in the development data!)

- **3D) Advebials**
  4 binary features:
  - *advmod*(TV, *) - presence of an adverbial modifier
  - *advcl*(TV, *) - presence of an adverbial clause modifier
  - *purpcl*(TV, *) - presence of a purpose clause modifier
  - *tmod*(TV, *) - presence of a temporal modifier

  And 3 categorical features – take the preposition p as a categorical value:
  - *prep*(TV, p) - presence of a prepositional modifier
  - *prepc_*p(TV, *) - presence of a prepositional clausal modifier

- *mark*(TV, p) - presence of a marker (= a subordinating conjunction different from "that" or "whether")

II. Semantic features

We will observe if a subject or an object of the target verb is a member of some of the 50 defined semantic classes, which have been derived from the EuroWordNet (for details see the Appendix C). The members (nouns) of the semantic classes are listed in the attached file data/semantic-classes.wn.txt. You will simply test whether some of the context words found in the sentence are listed under semantic classes or not.

The total number of the default (binary) semantic features is 200:

- presence of a nominal subject that belongs to a semantic class (50)
- presence of a nominal object that belongs to a semantic class (50)
- presence of a noun left of the TV that belongs to a semantic class (50)
- presence of a noun right of the TV that belongs to a semantic class (50).

5. Two steps and two deadlines

You will be given the data on November 23rd. In the first step, each student will be assigned one of three standard methods, namely Decision Trees or Naive Bayes or k-th Nearest Neighbour classifier. So you will have no choice of the method. Your task will be to apply the given method and to tune its parameters. Then you need to prepare a short report in the form of a written one-page description and an oral presentation.

Before the presentation, which will take place at the lab session on Friday, December 16th, you will send us your short report by the **first deadline, which is Wednesday, December 14th, 12pm**. You will need to turn in electronically to holub@ufal.mff.cuni.cz:

- **Your short report** (.pdf) describing methods, results and comments. The short report should be 1 page (A4) in length, excluding figures.
- **Your programming** (R codes).
- **Your slides** (.pdf) prepared for short oral presentation (6-8 minutes at maximum).

In the second step, you will solve both the basic task (A) and the advanced task (B). However, the choice of the methods is up to you. You must apply at least three machine learning algorithms from those you met during the lecture. All methods should be trained ONLY on the training data that you get. Reliable results should consist of information on the error rates expressed by the suitable measures (accuracy, confusion matrix) on your development test data. Comparison of the results on training and test data is welcome. Do not forget to compare results of different methods.

You do not have to present your final report publicly. Instead, you will defend your work individually. You should be able to explain all details and discuss the

choice of your solution in personal conversation with the teacher. **The deadline for your final report is Friday, February 17th, 12pm.** You will need to turn in electronically to holub@ufal.mff.cuni.cz:

- **Your final report** (.pdf) written according to the guidelines specified at http://ufal.mff.cuni.cz/~hladka/ML.html -> Projects.
- **Your final programming** (R code).

Filename convention

Everytime when you submit your work, please send always just *ONE zip file*, and follow the filename convention:

Whole package: "YourLastName.ml-project.2011-12.zip"

Your R-scripts inside the package: "YourLastName.ScriptNameofYourChoice.R"

Your report inside the package: "YourLastName.report.[short|final].pdf"

Remember well, that

your work MUST be done by February 17-th. After that deadline you cannot get "a signature". You can take the exam before you finish the final project. However, to get a final grade and the credit, you need to finish the project (i.e. to get "a signature").

Before you submit your final report you will have opportunity to consult Martin Holub about the problems you meet while working on the project. Do not hesitate to e-mail him to make appointment.

Appendix A

The Penn Treebank Tag Set

The tagset used in automatic morphological tagging is the Penn Treebank Tag set, described for example in Marcus et al. (1993). The following part-of-speech tags are used:

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Moreover, there are used the punctuation tags: [-LRB- | -RRB- | `` | '' | . | : | ,].

Appendix B

Stanford dependencies – example sentence

```
Savanna animals <cool> off with a kind of organic radiator by evaporating
water from the moist linings of the nasal chambers .

nn(animals-2, Savanna-1);
nsubj(cool-3, animals-2);
prt(cool-3, off-4);
det(kind-7, a-6);
prep_with(cool-3, kind-7);
amod(radiator-10, organic-9);
prep_of(kind-7, radiator-10);
prepc_by(cool-3, evaporating-12);
dobj(evaporating-12, water-13);
det(linings-17, the-15);
amod(linings-17, moist-16);
prep_from(evaporating-12, linings-17);
det(chambers-21, the-19);
amod(chambers-21, nasal-20);
prep_of(linings-17, chambers-21)
```

Appendix C

Semantic classes derived from EuroWordNet Top Ontology

The top ontology classification defined by Vossen et al. (1998) has been used to classify all words from WordNet into 50 semantic categories. For each word, we extract its hyperonymial hierarchy, which consists of a set of synsets. These synsets have been matched with categories of Vossen's top ontology. In most cases, the categories in Vossen's top ontology have the same names as synsets used in the WordNet, so we are able to match words with these categories immediately. However, there are also categories that do not match any WordNet synset directly. In these cases we use a few rules to map the synsets to the categories. The set of our rules has been proposed heuristically so that each word belongs at least to one of the categories.

The list of the rules used:

1. Synset "Abstraction" is mapped to category "3rdOrderEntity".

2. Synset "Phenomenon" is mapped to category "Phenomenal".

3. Synsets "Body of water" and "Matter" are mapped to category "Natural".

4. Synset "Process" is mapped to category "Dynamic".

These four rules are enough to get a list of words, where each word is mapped at least to one category. The list is given in the file data/semantic-classes.wn.txt.

References

- Silvie Cinková, Martin Holub, Pavel Rychlý, Lenka Smejkalová, Jana Šindlerová. 2010. Can Corpus Pattern Analysis Be Used in NLP? In *Text, Speech and Dialogue, 2010*. Berlin : Springer, 2010, pp. 67-74.
- Christiane Fellbaum, Joachim Grabowski, and Shari Landes. 1997. Analysis of a hand-tagging task. In *Proceedings of the ACL/Siglex Workshop*, NJ.
- Patrick Hanks and James Pustejovsky. 2005. A pattern dictionary for natural language processing. *Revue Francaise de linguistique applique*, 10(2).
- Adam Kilgarriff. 1997. ”I don't believe in word senses”. *Computers and the Humanities*, 31(2):91–113.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank, in *Computational Linguistics*, Vol. 19, No. 2, pp. 313--330 (Special Issue on Using Large Corpora).
- P. Vossen, L. Bloksma, H. Rodriquez, S. Climent, N. Calzolari, A. Roventini, F. Bertagna, A. Alonge, W. Peters. 1998. The EuroWordNet Base Concepts and Top Ontology. EuroWordNet (LE-4003) Deliverable D017D034D036, University of Amsterdam. http://www.vossen.info/docs/1998/D017.pdf.