Introduction to Machine Learning

Term project specification

PFL054 2012/13

Word Sense Disambiguation

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

1. What is the Word Sense Disambiguation task?

The classic task of W*ord Sense Disambiguation* (WSD) means to automatically assign the ''correct'' sense to a *target word* occurring in a given particular context. 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). A brief overview of the WSD field is available at Wikipedia (see section References).

Motivation

WSD is a possible approach to *lexical disambiguation*, which 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 3 target words:

- *line* (noun),
- *hard* (adjective),
- *serve* (verb).

For each target word you will implement a supervised classifier. As your training (development) data you will use a set of manually annotated sentences for each target word, which are available in the directory `data/*.devel`. This data was developed by Leacock et al. and used in numerous comparative studies of word sense disambiguation methodologies. In brief, each instance has been tagged with one of WordNet senses. Further details can be found in the paper (Leacock et al.).

Each sentence 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.

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). 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.

3. Data description

Primary data

The manually annotated data sets are stored in 3 text files (each file with a number of sentences containing the same target word) in the directory `data/*.devel`. Please, use the provided data only for your study or academic purposes. You are not allowed to distribute it.

Each data instance consists of 6 lines/items:

- *sentence ID* – you do not need it;
- *sense identification* – which is, in fact, the manually annotated *class label*;
- *tokenized sentence* with marked *target word* – tokens are separated by spaces;
- *morphologically analysed sentence* (MORPH) – morphological tags have been determined automatically (thus, some errors can occur); the description of morphological tags is available in the Appendix A;
- *morphologically analysed sentence* (WLT) – an alternative format including lemmas; 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 by an automatic tagger different from the previous one (some differences can occur);
- list of *syntactic dependencies* (PARSE) – 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).

Development set and test set

You will get a number of manually classified instances for each target word that make your development data sets. There are another 500 instances for each target word that make *test sets* ("unseen data 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. The following table summarizes the number of all available data instances.

target word	data instances		
	training	test	total
line	3646	500	4146
hard	3833	500	4333
serve	3878	500	4378

Our *recommendation* is to split your (development) data (for each target word) into 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 instances you have got) to train your final, "best" classifiers. Those final classifiers will be submitted and then evaluated on the unseen test sets.

Feature extraction

Each data instance to be classified consists of the *target word* and some context words. Therefore the values of the features that describe data instances can be based on the observed characteristics of both the target word or the context words. All features are either numerical, or binary (T/F values), or categorical (listed, discrete, non-numerical values). You can use anything what you can find and extract from the given sentences and make your own feature vectors.

4. Two steps and two deadlines

You will be given the data on November 30th. In the first step, each student will be assigned one standard machine learning method. So you will have no choice of the method. Your task will be to extract your feature set and then to apply the given method, to tune its parameters, and to evaluate its performance. 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 21st, you will send us your short report by the **first deadline, which is Wednesday, December 19th, 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** (codes for feature extraction and R codes).
- **Your slides** (.pdf) prepared for short oral presentation (6-8 minutes at maximum).

The final report

In the second step, the choice of machine learning 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 matrices) on your development test data. Comparison of the results on (development) training and (development) 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 22nd, 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** (codes for feature extraction and R codes).

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.2012-13.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 22nd. 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.

5. References

- Christiane Fellbaum, Joachim Grabowski, and Shari Landes. 1997. Analysis of a hand-tagging task. In *Proceedings of the ACL/Siglex Workshop*, NJ.

- 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).

- http://en.wikipedia.org/wiki/Word-sense_disambiguation.

- Leacock, Chodorow and Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* 24:1.

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

Example feature extraction: Morpho-syntactic features useful for verbal target words

As a hint for your feature extraction, here are 83 morpho-syntactic features suggested. 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 target verb (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")