Introduction to Machine Learning

Term project specification

PFL054 2010/11

Semantic Collocation Recognition

The aim of the term project is to solve "the best you can" binary classification task. Your classifier will work with word pairs and should recognize semantic collocations: it should distinguish between them and word pairs that do not form semantic collocation.

What are semantic collocations?

If one word collocates with another, they often occur together. Most generally, the term *collocation* denotes a meaningful and grammatical word combination that often (regularly or frequently or typically) occurs in natural language.

Semantic collocations, in addition, form semantic units. Semantic collocations are multiword expressions that are lexically, syntactically, semantically, pragmatically and/or statistically idiosyncratic. It means that semantic collocations have semantic and/or syntactic properties that cannot be fully predicted from those of their components, and therefore semantic collocations have to be listed in a lexicon.

For example, compare the following 5 expressions: "they are", "black horse", "red wine", "black market", and "weapons of mass destruction". While the first two are considered only "simple" collocations, the other three are semantic collocations, according to our definition.

Motivation

Collocation extraction is a traditional task of corpus linguistics. The goal is to extract a list of semantic collocations from a text corpus, which then makes a collocation lexicon. The knowledge of semantic collocations is important and can be used in many subfields of applied computational linguistics, e.g. in computational lexicography, natural language generation, word sense disambiguation, machine translation, information retrieval, or identification of technical terminology.

Your task in detail

In this project, you will deal only with two word Czech semantic collocations (henceforth simply *collocations*). As your training data you will get a set of manually annotated collocation candidates. Each candidate is a *bigram* (i.e. a word pair), manually classified either as collocation or as non-collocation. The following table shows a dozen of collocation candidates:

example	**translation**	**semantic collocation**
další jednání	further negotiation	no
první republika	First Republic	yes
metr čtvereční	square meter	yes
zahraniční banka	foreign bank	no
nový ředitel	new director	no
znovu potvrdit	to confirm again	no
doba splatnosti	term of expiration	yes
poslední měsíc	last month	no
trestný čin	criminal act	yes
kroutit hlavou	shake head	yes
dobrý tenista	good tennis player	no
velmi důležitý	very important	no

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 classifier 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. In case you do not have just one classifier that clearly outperforms the others, you are allowed to propose two of them.

Your best classifier will be evaluated on our test set (that will be hidden from you until you submit your final classifier). In case you propose two classifiers, both will be evaluated. The student with best results on our test data will be the winner.

Data

All annotated collocation candidates have been extracted from a Czech 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 annotated data set is stored and described in attached files candidates.train.frequency.9232.csv and candidates.train.features.9232.csv. The format of the files is described in file candidates.format.txt. You will find three kinds of information about each candidate:

1) *Basic statistical characteristics* in the form of four frequency values attached to each candidate. All frequencies were counted in the same corpus. Candidates are described as word lemma pairs (x, y), and the four frequency values are

A = count(x, y) ... frequency of the candidate in the corpus;
B = count(¬x, y) ... frequency of bigrams in the corpus in which a word different from word x is followed by word y;
C = count(x, ¬y) ... frequency of bigrams in the corpus in which word x is followed by a word different from word y;
D = count(¬x, ¬y) ... frequency of bigrams in the corpus in which a word different from word x is followed by a word different from word y.

In fact, these four values form the contingency table of the frequencies associated with the bigram x y.

2) *Feature vectors* of 82 numeric variables. All of those variables are numeric continuous values and are computed from the basic values A, B, C, and D. The complete list of these variables with exact formulas is given in file features.description.pdf.

3) *Manually annotated class labels.* Each label has the value of 1 (collocation), or 0 (non-collocation). Use the feature named "tp" (the last one) as the class label.

Training set and test set

You will get 9,232 such manually classified instances. There are another 3,000 instances that make a *test set*, which you cannot see until you finish your "best classifier" and submit your final report. Then your classifier will be evaluated using the test set.

Our recommendation is to split your data in two parts, a *development training set* and a *development test set*. When you develop your classifier, you will use the development test set both to evaluate your classifier/s and to tune its/their parameters. Setting a suitable number of the features used for classification is a part of the project. Once you have finished your parameters tuning, you will choose the best model and use all the annotated training data (i.e. all 9,232 instances you have got) to train your final, "best" classifier. That final classifier will be submitted and then evaluated on the unseen test set (the unseen 3,000 instances).

Two steps and two deadlines

1) You will be given the data on November 10th. 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 oral presentation.

Before the presentation, which will take place at the lab session on Wednesday, December 15th, you will send us your short report by the **first deadline, which is Wednesday, December 8th, 24:00**. 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 (5-10 minutes at maximum).

2) In the second step, you will solve the same task. However, the choice of the methods is up to you. You must apply at least three machine learning algorithms

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, precision, recall, F-measure) 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 Sunday, February 20th, 24:00.** 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.2010-11.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 20-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 final project (I.e. to have “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.