LIN3011/LIN3012 Data-Driven NLP

Final Projects

Instructions

Choose one of the projects described below. Each project may be done individually or in a group and delivered through the VLE upload area in the form of a write-up, by the deadline indicated on the VLE.

If the project is done in a group, you need to inform Albert in advance about who the group members are. Groups cannot consist of more than three people.

Layout

Your assignment needs to incorporate sections on the following:

- Relevant literature review
- Data and Methodology
- Results
- Discussion and conclusions

Length

The length for assignments is as follows:

- Individual projects: between 15 and 17 pages, double-spaced, 12pt font
- Group projects: 20 to 25 pages, double-spaced, 12pt font

The above does **not** include references.

Project 1: Classification and WSD

In this project, you will work with the sense-tagged data for the word *hard* provided in file *hard.POS*. This file contains instances of the word in context and is distributed with the python Natural Language ToolKit. Note that the data is in XML format.

Your task is to develop classifiers which, given an instance of the word *hard* in context, determines which of its senses is the intended one. In particular, you should:

1. Develop a core set of features to classify the sense of the word *hard* in context. These may include the words or parts of speech of the word within a particular window (e.g. two words to the left and/or right), as well as any other features that you consider relevant based on your literature review;
2. Compare the performance of two classification algorithms (e.g. Naive Bayes and MaxEnt) in the performance of the word sense disambiguation task;
3. Compare these algorithms against a baseline;
4. If necessary, perform feature ablation to determine which are the best features to use.

Your report should details the results of a thorough evaluation using a cross-validation methodology, for both algorithms used and the baseline. Your discussion should seek to explain the trends discovered in your results (for example, why certain methods perform better than others).

You get bonus points for using additional features in addition to what was done in class, and for varying the windows for these contextual features (i.e. conducting experiments using, say, two versus three context/word tags as features).

Useful resources:

You are free to use existing libraries or implementations of classifiers.

- Should you decide to work with NLTK, you may wish to consult the how-to pages on classification in the NLTK book (Ch. 6): http://nltk.org/book/ch06.html
- WEKA contains JAVA implementations of classification algorithms, including all of those discussed in class.

Project 2: Blog gender identification

This project is concerned with the use of statistical models or classifiers to identify the gender of authors of blogs. Work in this area has identified a number of interesting variables in people's use of language that helps to identify them. Examples include their use of function words, the hapax legomena in their text, etc.

Your aim in this project is therefore to identify the linguistic markers of gender, justifying your choice of features with reference to the literature, and applying a machine-learning methodology to conduct your experiments.

The steps involved are as follows:

1. Find blog texts written by different authors of different genders. You can use the corpus linked below, or a sample of it.
2. Build and train a model which, given an input text by one of the authors, extracts its features and classifies it according to the most likely author gender. Note that it is possible to view this as a classification task. It is up to you to choose the classification algorithm to deploy. You should, however, compare your results to an appropriate baseline.
3. Evaluate the model using held-out test data or using a cross-validation design.

Useful resources:

- The Blog Authorship Corpus, constructed by Moshe Koppel, is a very large corpus of blog posts by multiple authors, with some demographic variables about authors available (e.g. age, gender and astrological sign)
- This paper describes some experiments on this corpus: J. Schler, M. Koppel, S. Argamon and J. Pennebaker (2006). Effects of age and gender on blogging. *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*

Project 3: Part of speech tagging

The task for this project is to design and implement a probabilistic part of speech tagger. **The project description below is based on constructing a tagger for Maltese. However, you are also free to conduct the same study on a different language**. If you decide to do so, please contact Albert in advance and let him know as it is important to have appropriate pre-tagged data for this study.

There currently exists a POS-tagged corpus of Maltese, divided into several sections. This corpus is called *Korpus Malti v3.*0. The Corpus itself can be browsed online, or downloaded. It is divided into several sections. For this project, you are advised to **use the "newspaper" section** of the corpus. (See below for details).

The corpus texts are in a vertical format, as shown in the following example:

Fl-	PREP-DEF	fi	null
aħħar	NOUN	aħħar	null
mill-	PREP-DEF	minn	null
aħħar	NOUN	aħħar	null
,	X-PUN	,	null
kitbu	VERB	kiteb	k-t-b
li	COMP	li	null
l-	DEF	il-	null
għan	NOUN	għan	null
ta'	GEN	ta'	null
CMM	X-ABV	CMM	null
huwa	PRON-PERS	huwa	null
li	COMP	li	null
joffri	VERB	offra	null
attivitajiet	NOUN	attività	null

(This example would be translated as: *At the end of the day, they wrote that the aim of CMM is to offer activities*)

In the above, the first column is the actual word used. The second column is the POS Tag, the third is the lemma (the morphological base form, for example, in the case of *attivitajiet*, which is a plural noun, the base form is the singular; similarly, for *kitbu*, the base form is *kiteb*). The final column is the root (which is null for all words except those of Arabic origin, which have a consonantal root, as in the case of *kitbu* above, whose root is *k-t-b*).

You should treat this corpus as your training and test data, i..e you will be developing a new POS Tagger and comparing the outcomes against the corpus.

Here is what you minimally need to do:

1. Train your tagger.
2. Test the tagger on held-out test data.
3. Evaluate the results. This should also include an error analysis, i.e. a discussion of where the tagger goes wrong.

You are free to choose any technique for training your POS tagger (e.g. HMM, MEMM etc). The features you use are also up to you. In most instances, you'll be using contextual features, but in some cases (as in the case of a MEMM) you are able to also use word-level features (such as word endings).

Useful resources:

- The Maltese Corpus is available on the Maltese Language Resource Server: http://mlrs.research.um.edu.mt. If you visit the "Downloads" section of this page, you will be able to download individual sections of the corpus, including the newspaper section.
- The Part of Speech Tagset developed for Maltese and used in this corpus is described here: http://mlrs.research.um.edu.mt/resources/malti03/tagset30.html

Project 4: Stylistically constrained sentence generation

In this project, you will be generating sentences based on an n-gram model. Your aim is to model the style of a particular author.

You need to collect two corpora. Each corpus will be from a different author (e.g. Jane Austen and Charles Dickens). Note that you don't need to focus purely on a literary author – you could just as easily try to model an author who wrote academic text (e.g. based on a corpus of their papers), or an author who wrote several blogs.

Once you have collected the corpora, you will need to build a language model representing each author. It is up to you to decide the length of n-grams you will consider, and the smoothing technique(s) you choose. However, your choices should be motivated based on a perplexity evaluation.

Then, once you have built your models, write a program that will generate sentences in the style of that author (i.e. using the probabilities from the language model for the author).

Finally, run a **human** evaluation, where you ask a small number of people to judge the fluency and readability of a fixed set of generated sentences. The best way to do this is to show people some sample sentences, and ask for fluency judgements on a Likert-type scale (where one gives a rating from, say, 1 (not fluent at all) to 5 (very fluent)). You may do this any way you see fit (e.g. online, by contacting friends, etc). The aim of this is to get a sense of how good the random generation is.

Your write-up should include examples of the generated text.

You are free to carry out this project in any language you choose.

Useful resources:

Widely-used language modelling toolkits include the following:

- The SRI language modeling toolkit: http://www.speech.sri.com/projects/srilm/
- The CMU-Cambridge language modeling toolkit: http://www.speech.cs.cmu.edu/SLM/toolkit.html
- NLTK's built-in libraries for langauge modelling

Project 5: Sentiment Analysis of Facebook Posts

Sentiment analysis is the task of predicting the affect (emotion) in a text. For example, a movie review might be classified as overall positive or overall negative, depending on the types of expressions used. In a similar vein, social media posts, such as the posts on someone's Facebook wall or a person's tweets, often have emotional content.

In this project, you will use a large dataset of Facebook posts, collected as part of the MyPersonality project, to build a statistical model that predicts the emotion (positive or negative) in a Facebook post.

The data available map a number of FB posts to two classes, one representing Valence/Sentiment (positive, negative) and one representing Arousal (which can range from low to high). In the dataset, you are given two judgements of Valence and two judgements of Arousal per post. They are represented numerically. The details of the data are available in a paper linked below.

Your task is:

1. To determine a set of features of a text that can predict Valence and Arousal (separately or jointly);
2. To build a model which, given a facebook post, will predict its valence and arousal based on these features – you can treat this as a classification problem (hence, using any classifier you deem fit) or as a numerical prediction (perhaps modelling this as a regression problem);
3. To evaluate the model using an appropriate methodology (via held-out test data or using cross-validation) and against a baseline.

Useful resources

- The MyPErsonality project is fully described on the following page: http://mypersonality.org/wiki/doku.php?id=download_databases. The data relevant for this project is available to unregistered users from this site (see the heading *Status updates annotated with valence and arousal*).
- The following paper describes the annotation of the sentiment data from MyPersonality: https://sites.sas.upenn.edu/danielpr/files/va16wassa.pdf
- An overview of the task of sentiment analysis can be found in this paper by Bo Pang and Lillian Lee: http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf