LIN3011/LIN3012 Data-Driven NLP

Final Projects

Instructions

Choose **one** of the projects described below. Each project should be delivered through the VLE upload area in the form of a write-up and associated code, by the deadline indicated on the VLE.

Projects need to be done individually.

The deadline for submission is February 9th, 2020, at 23:59.

Layout

Your assignment **must** incorporate sections on the following:

1. An introductory section, giving a problem definition and an appropriate motivation.
2. A brief review of relevant literature, showing that you have familiarised yourself with related work that seeks to solve this, or similar, problems.
3. Data and Methodology. This is the most important section. It must include:
    a. A full description of the data used, including any train/dev/test splits and how they were determined.
    b. A description of the methods used, with details of architecture, hyperparameters etc.
4. Results, presented clearly using tables, figures or any other appropriate media.
    a. Where possible, compare your results directly with existing state of the art.
    b. If you have used a baseline, be sure to compare your main models with the baselines.
    c. Where possible, in addition to figures on performance, perform a short qualitative analysis, showing examples of outputs accompanied by a discussion of where things went wrong (and where they went well).
5. Discussion and conclusions

Code and data

- Ony submit data if it isn't a standard release, Ie. If you created it yourself from scratch or by modifying an existing dataset. If you just used an existing dataset, you don't need to submit it, but include a reference to the relevant work describing it, plus a link if relevant.
- Code can be written in a programming language of your choice. Ensure that code you submit includes:
    - Brief but clear documentation
    - A README file that allows one to run it

- Code and data should be submitted together as a zipped archive through VLE, or with a link from an online repository that can be cloned via git. No other formats will be accepted.

Length

Project reports should be between 8 and 10 pages, double-spaced, 12pt font.

The above length limit does **not** include bibliography.

Assessment criteria

Projects will be assessed based on the following criteria:

- **Methodology (50%)**: Is your method appropriate for the chosen topic? Did you train and test your models properly and are you reporting appropriate baselines for comparison?
- **Presentation and write-up (25%):** Does your write-up incorporate an appropriate literature review, data/methods presentation and discussion? Do you present your results in a transparent form, using tables, graphs etc? If you conducted some form of supervised learning, do you describe your features appropriately?
- **Coding or data collection effort (25%):** If you submit code, is it well-commented? Does it actually do what it says on the tin? If you've collected additional data, does your report describe it properly and is it clear what you've collected and why?

Note that I do not accept project write-ups which are basically just code documentation. This is not primarily a coding project. Your aims are scientific – your code is a tool to achieve those aims. Your write-up is a scientific report, not a manual.

Project 1: 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. Identify gender-relevant features.
3. 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.
4. Evaluate the model using test data or using a cross-validation design.

Useful resources:

1. 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)
2. An oldish paper to get you started (but you'll need to go further than this): 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 2: Stylistically controlled sentence generation

In this project, you will be generating texts based on stylistic features. The dataset used was developed by Ficler and Goldberg (2017). It is a set of movie reviews harvested from the website rottentomatoes.com .

Ficler and Goldberg (2017) are interested in controlling linguistic style in automatically generated text. Specifically, they are interested in modelling the features of a sentence in a review that determine, among other things:

1. Whether it is professionally written by a critic, or by an amateur;
2. Which aspect of a movie (e.g. plot, acting) the sentence describes;
3. Whether the sentence is descriptive;
4. Whether the sentence is written in a personal ("I like this…") or objective ("This movie is…") style;
5. The overall sentiment of the review, based on the rating given by the critic.

Your task in this project is to design and implement a model that generates sentences. The model should be defined as follows:

- INPUT: A set of parameter values according to the five features above
- OUTPUT: A sentence in English that reflects the parameter values.

In this project, you are **strongly encouraged** to also experiment with additional stylistic features, over and above what Ficler and Goldberg used.

Useful resources:

The paper from which this data is obtained:

- Ficler, J., & Goldberg, Y. (2017). Controlling Linguistic Style Aspects in Neural Language Generation. In *Proceedings of the Workshop on Stylistic Variation* (pp. 94–104). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from http://arxiv.org/abs/1707.02633
- **NB:** THe supplementary material contains hints about how to annotate each sentence automatically.
- The data is linked from the VLE. It is provided to you with the kind permission of Jessica Ficler. **Do NOT distribute this data.**

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

- There are plenty of libraries for training and testing recurrent neural nets, if you decde to implement a neural model. You get additional credit for modelling this using a neural network language model, as Ficler and Goldberg in fact did.

Project 3: Multilingual EMOJI prediction

Emojis are part and parcel of people's interactions on social media, where they convey everything from emotion to content that can be concisely expressed in a non-purely linguistic form.

From a linguistic perspective, it seems reasonable to predict that the use of an emoji (say, a heart) in the context of a longer message, such as a tweet, is somehow connected with the content of that message.

For this project, you will be experimenting with a task that was conducted as a shared task at the SemEval 2018 conference. The setup of this task was simple:

- Data: 500k tweets in English, and 100k tweets in Spanish, each of which contains exactly one emoji.
- THe emojis fall into one of exactly 20 possible classes, which function as the labels in this task.
- The aim is, given a tweet, to predict which emoji is the correct one.

Useful resources:

A full description of the Shared task, including a summary of the results by various participating systems, can be found here: https://competitions.codalab.org/competitions/17344#learn_the_details-data

Since this is a shared task, the data, including the test data, is all available on the above URL.

Note that the task organisers also provide an evaluation script. Feel free to use this, but be sure to understand what it does, and include details of this, and any other evaluation method you decide to use, in your report.

Useful literature:

- The shared task results are described in this paper by Barbieri et al (2018): https://www.aclweb.org/anthology/S18-1003/
- Useful background on emoji prediction can be found in this paper: https://arxiv.org/abs/1702.07285

Project 4: Language Identification

Language identification is the process of automatically determining which language a piece of text is written in. Typically, this is framed as a classification problem, where given a string and a finite set of language classes (e.g. EN, DE, MT, ES, etc) the aim is to identify the most likely language for the string.

This problem is usually solved using techniques involving character distributions (since certain character sequences are more likely in some languages than in others). For example, the sequence *g-ħ* corresponds to the letter 'għ' in Maltese, which isn't found in most other European languages.

For this project, you will be comparing different techniques for language identification. In particular, you are encouraged to experiment with:

- Different types of features (especially character ngrams of different lengths)
- Different classifiers, in particular, log-linear versus neural/MLP classifiers.
- For the specific case of Maltese (which is included in your data – see below), you should also consider the problem of diacritics: in Maltese, the characters *ħ*, *ġ* etc are good signals for the right label, but it is also true that in many contexts, people omit them. Can your classifier achieve good results even when the strings do not include diacritics?

Useful resources:

The data for this task can be drawn from data made available by the European Parliament, which consists of EU documents in all the official EU langauges. This means you can build a classifier to distinguish between 24 possible languages. The corpus is called the Digital Corpus of the European Parliament. It is available here.

Useful literature:

There is a lot of literature in language identification. A good reference to start with, which also describes an excellent system (called langid.py) is this paper.

Project 5: Bias detection using distributional models

Bias is pervasive in language and recent work suggests that bias can be detected using word embeddings. For example, Bolukbasi et al (2016) show that certain words are more strongly associated with certain genders. Similarly, Garg et al (2017) show that racial and gender bias revealed in word embeddings is linked to social and cultural change.

Most of this work has been done on English. In this project, you will explore gender bias in a different language (Maltese, or another language of your choice, but not English). You will need to:

- Collect or find a corpus in the language you have chosen. (E.g. if you are using Maltese, you could use the Korpus Malti v3.0, available on the Maltese Language Resource Server)
- Build a distributional semantic model. In particular, you are encouraged to experiment with more than one state of the art embedding model, such as Word2Vec (Mikolov et al, 2013) and GloVE (Pennington et al, 2014). You are further encouraged to compare the outcomes of these embeddings with a more "traditional" distributional semantic model, based on counts rather than predictions.
- Bolukbasi et al (2016) use core 'seed' terms which are naturally gendered (e.g. words like *sister* and *brother* are obviously female and male), as well as pronouns. They treat these as the points around whcih they explore the clustering of other terms. Adopt an approach similar to this one (but adapted to the language of your choice). As you do this, focus in particular on:
    - Nouns which denote occupations (such as *doctor, nurse, teacher*)
    - Adjectives (*such as nice, strong, sweet*)
- Explore methods of de-biasing your embeddings. Give evidence that de-biasing helps in maintaining core semantic relationships, while removing the kinds of biases you have explored.

Useful literature:

Bolukbasi, T., Chang, K., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS'16)* (pp. 1–9). Barcelona, Spain.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186. https://doi.org/10.1126/science.aal4230

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2017). Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644. https://doi.org/10.1073/pnas.1720347115