Assignment 1

COMP550 Natural Language Processing- fall 2019

Badreddine SAYAH – ID - 260940022

Question 1

(apologies for the intermediate English level, I am not a native English speaker)

Every student took a course.

The sentence has at least two meanings or two logical forms:

For every student there is a particular course.

There is at least one course for every student.

This is caused by the scope of the quantifier (and the determiner) *"every"* [1].

The ambiguity here is syntactic, specifically quantifier scope.

Moreover, we can consider that the word "course" may have two senses at least [2], the first one is school or university course, and second meaning could be "the direction". Thus, this may involve semantic ambiguity even if in the sentence we are talking about student: *Every student took a direction.*

To disambiguate the passage and let us consider an interpretation more than another, we need more information with regard to the scope of the quantifier, or even specifying a number.

John was upset at Kevin but he didn't care.

We have an ambiguity in these passage related to the pronoun "he". So we can have two possibilities:

John was upset at Kevin but he(=Kevin) didn't care.
John was upset at Kevin but he(=John) didn't care.

This type of ambiguity is Discourse, anaphoric ambiguity

A textual coherence or narrative concept can resolve the ambiguity in this passage. Thus, some additional information about the behavior and reactions of the two-person cited in the sentence may bring more clarifications and more sense to choose the person related to the pronoun "He" in the passage.

Moreover, we can consider the recency rule, cited in [3] as "the referent is the most recently mentioned object of correct gender and number"

Sara owns the newspaper.

this passage has two meanings:

Sara is the owner of the newspaper, the newspaper as a company.

Or, Sara owns the newspaper, as she regularly read it or she might be the first one to read it and monopolize time of reading it so, every person has to wait for his turn.

We can talk about the literal and figurative meaning but even the literal one is not clear in this case. The ambiguity is caused by the expression it self "owns the newspaper", but also the word "the newspaper" which can have two meaning as we said before.

This is a semantic ambiguity.

The Sort of knowledge needed to disambiguate this sentence can be the speaker intent, when the speaker wants to communicate a specific idea or information about Sara, for example if she is the owner of the company newspaper, can be communicated with certain intent. Whereas the second meaning could have another tone. Also, it could depend on the context, if for example in the text or in the conversation; a regular habit is mentioned, like reading the newspaper for a long time…

He is my ex-father-in-law-to-be.

the hyphened term "**ex-father-in-law-to-be"** is causing ambiguity. There is the informal noun "ex" and everything else; considering each word a part we may have other meaning like for "law" and to-be.

Ex-father-in-law-to-be, can mean that the speaker is going to divorce, or she/he is engaged, and all is cancelled. Another possibility is the speaker has started the divorce procedure: if we look to the part "to-be" equivalent to the future.

The ambiguity here could be structural, as well as semantic.

Non-linguistic knowledge to understand the situation and resolve the ambiguity of the passage.

ttyl ;)

the abbreviation ttyl = talk to you later, abbreviations in text and chat message is hard even for human.

Later = not specific, minutes, hours, days…

The emoji ;) is ambiguous too; following its definition or description [4]: "May signal a joke, flirtation, hidden meaning, or general positivity. Tone varies, including playful, affectionate, suggestive, or ironic. "

;) Can be considered as punctuation too, so the character semicolon ";" is used to separate two independent but related clauses: The semicolon is also used to separate list items when the list items contain commas [5], however the closing bracket can cancel this idea.

Once we define the possible expansion of ttyl : “talk to you later”, this expression contain an ambiguity that could be considered of Pragmatic type. Might be lexical too in the case of ttyl= text you later, as a possibility.

Some additional knowledge like how often they talk can bring some clarification about the exact meaning of “later” in this passage. In the case we are interested in the “exact” meaning of the emoji, the text of the conversation or other conversations between “this two persons” can bring more information if these persons are flirting or joking or..

Questions 2

In this question we show that Naive Bayes classifier is a linear classifier over its feature space when using categorical distributions for the features

For simplicity, let’s start with the case of two categories. These two categories can be coded in one-hot manner, so:

$$p(Y=1|X) = \frac{p(X|Y=1)\,p(Y=1)}{p(X|Y=1)p(Y=1)+p(X|Y=0)p(Y=0)}$$

Dividing by the denominator we have:

$$p(Y=1|X) = \frac{1}{1+\frac{p(X|Y=0)p(Y=0)}{p(X|Y=1)p(Y=1)}}$$

$$= \frac{1}{1+\exp\left(-\log\frac{p(X|Y=0)\,p(Y=0)}{p(X|Y=1)\,p(Y=1)}\right)}$$

With the conditional independence assumption, we get:

$$p(Y=1|X) = \frac{1}{1+\exp\left(\log\frac{p(Y=0)}{p(Y=1)} + \sum\log\frac{p(X|Y=0)}{p(X|Y=1)}\ \right)} \qquad (1)$$

The second part of the denominator is in a linear form $(\vec{w}^T\vec{x} = b)$ or: $w_i x_i + b$ (in the log space) with w_i = $\log\frac{p(X|Y=0)}{p(X|Y=1)}$ x_i where x_i is the number of occurrence of X. and $b = \frac{p(Y=0)}{p(Y=1)}$

We can see also that the equation (1) it is in logistic regression form. Additionally, given the canonical form of exponential family distribution (categorical, Gaussian...) [8]:

$$f_x(x|\theta) = h(x)\exp\ (\eta\,(\theta).\mathrm{T}\,(x) - A\,(\eta))$$

So we can rewrite (1) as:

$$\boldsymbol{p(Y = 1|X) = \ \sigma\,(\textstyle\sum_i log\,\frac{p(X|Y=1\,)}{p(X|Y=0\,)} + log\,\frac{p(Y=1\,)}{p(Y=0\,)})}$$

$\boldsymbol{\sigma}$ is the logistic function. And this can be generalized to multiple categories.

Questions 3

In this question we compare classifiers for a supervised text classification problem: sentiments analysis. The goal is to classify a given a sentence or a snippet extracted from a movie review dataset, as positive or negative comment. Moreover, we investigate most common techniques of text analysis and normalization.

The snippets are stored in two separated files (each one has 5331 lines), labelled as positive and negative reviews. We concatenate them in a numpy ndarray and (one-hot) code the corresponding labels; 1 for positive sentences, and 0 for negative ones. By doing preliminary visual exploration we found at least two sentences in a foreign language, and some joined words with and without hyphens.

Pre-processing

We extracted feature vector representation as unigram counts, using countVectorizer module of scikit learn. we explored whether to use English stop words lists or not, two lists were investigated from nltk corpus and the second sklearn. We compared this with tuning threshold for removing the most frequent words in the texts.

For stemming we used PorterStemmer. LuncasterStemmer() was explored too but seemed to be over-stemming and slower than PorterStemmer. We used WordNet lemmatizer for which we needed to determine a pos-tag for each token automatically; we run the nltk post tag on a tokenized sentence, and pass it ((as compatible word-net or hashable type) to the WordnetLemmatizer [11]. finally, we splitted the data or the snippets in a train and test sets, 80% and 20 % respectively.

Experiments

A dummy classifier (sklearn) has been used as baseline, for which two strategies were compared, stratified and uniform, the latter was better. The other classifiers have been tuned separately. For the logistic regression (LR) we've tried different penalties and solvers. We've search for the ideal values of C (inverse of regularization strength for LR) in a range of [10^{-4} , 50, for linear SVM (where C='penalty'). Moreover, two versions of the naive Bayes classifier were investigated, Multinomial and Bernoulli for which the best smoothing parameter alpha was obtain from a parameter range [10^{-2} , 10].

We've tuned the threshold for removing infrequent words within a uniform range [10^{-2} , 1] which improved slightly all classifiers accuracy, with an approximative mean of 0.5% (best min_df and max_df differs from a classifier to another, see code). We gain another 0.5% using Tokenizer built with a different pattern (r'[a-zA-Z0-9]+') [10]. LR is the most classifier that benefits from this, with an accuracy from to 77.16 % (stemming) and 76.79 % (lemmatizations) to 77.87 % and 77.16 %.

Results and discussion

Using two different stop word lists decreased similarly the performance, independently of the other parameters. Thus, tuning the threshold for removing the most frequent words (max_df) was a better choice. Lemmatization and stemming improved both the results. But when compared to each other, stemming gave a slightly better results with LR and linear SVM (Figure. 2), Even if it is known that lemmatization is more sophisticated by producing most suitable inflection for the language.

All classifiers have performed better than the baseline; the LR with stemming is the best classifier with an accuracy of 77.91%, even if the Bernoulli naïve Bayes gave 78.15% (which is not reported here, because we believe it's not suitable for this task). Finally, as we can see in the confusion matrix (Figure. 1) the classifier misclassified positive sentences (false negatives = 262) more than negative ones (false positives= 209). With more attentive inspection we found that some snippets classified as positive, uses irony or sarcasm.

	Positive	*Negative*
Positive	834	262
Negative	209	828

Figure 1. Confusion Matrix (LR)

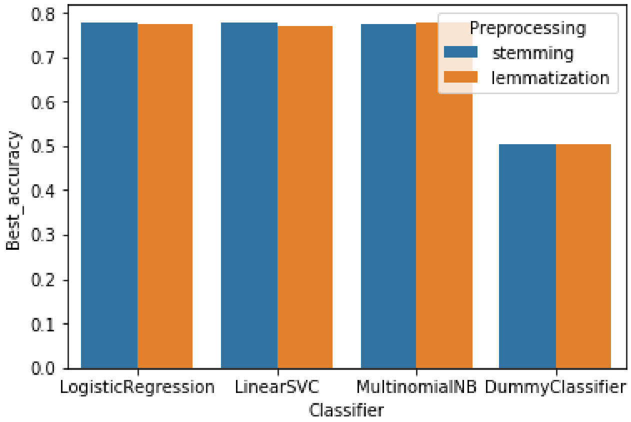

Figure 2. Classifiers accuracy with different pre-processing

References:

[1] Ambiguity, Stanford Encyclopedia of Philosophy

[2] macmillandictionary.com/course

[3] Notes on ambiguity

[4] emojipedia.org/winking-face/

[5] wiki/Punctuation_of_English

[6] Jufraski, Speech and language processing, 2nd edition, pp. 24, 25, 29,156, 223, 349, 513.

[7] Tom M.Mitchell. Machine learning. Genrerative and discriminative classifiers.

[8] Exponential_family

[9] https://www.nltk.org/book/ch06.html

[10] https://www.machinelearningplus.com/nlp/lemmatization-examples-python/

[11] https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

Notes: these are hypertlinkd,