CS246 Progress Report: Twitter Purifier

Wen Shi, Zijun Xue, Jennifer Zhang

April 2014

Current Progress

We currently have harvested approximately 150,000 tweets via the Twitter sample API.

Building an abbreviation mapping list

Using a combination of UNIX's dictionary, Google Translate's list of most frequently used words[1], and the most common words from TV and movie scripts[2], the most frequent mismatched words from Twitter were found. Among these, around 50 commonly known abbreviations that were also phonetically far from their intended expansion were manually mapped. This process was made more difficult by proper nouns – particularly celebrities, names of companies, and sport teams. It may be desirable to improve this process by also crosschecking against a list of well-known public figures, but since this will not be affect the final product it may not be worthwhile.

Squeezing

Inserting extra duplicate characters in a word is called squeezing. We observe the squeezing phenomenon quite often on Twitter, e.g. *I am soooooo happy!*, *it's gooooooooooood*. Usually, people use this to express their emotions or to emphasize particular words. We developed a rule-based filter to "clean" such squeezed words based on the letter being duplicated. In our algorithm, we generate a candidate list for all possible original words since there are cases with multiple possible original words. For example, we can find tweets *Oh my gooooooooooood!* and *it's goooooooooooooood*. In such a situation, we cannot distinguish between "god" or "good" by looking at the original string, so we leave all possibilities in the candidate correction list. The module for this has been completed.

[1]google-10000-english
[2]Wiktionary Frequency Lists

Word frequency index

A word frequency dictionary "databaseDict.jason" has been created from the 150 000 tweets, in which words with special characters have been filtered out (usernames, hashtags, links, etc). This dictionary will be used for generating the candidate word list for an error words since we the word frequency will be used as the probability of a certain word in Bayesian function. From the dictionary, we see many of the "word"s only has a frequency of 1, we assumed that words with a frequency of 1 has a very high possibilities to be a spelling error and words with a high frequency (threshold of 3 4) will be a correct spelling.

Inverse phonetic index

The inverse phonetic index "soundexDict.jason" has been established based on the Soundex phonetic algorithm[3], which is for indexing words by sound, as pronounced in English. We indexed each of the words in the 150,000 tweets to its phonetic index and whenever we find a spelling error, we will assume that the corrected words might have the same phonetic index. Associated words will be part of the candidate corrections word list.

Changes

Ambiguous or rare abbreviations

In the course of mapping out abbreviations, it is clear that many abbreviations may have different intended expansions under different contexts. E.g. *hw* overwhelmingly represents *homework* as an abbreviation, but is also a typo for *how*. *kd* in an NBA related tweet likely refers to *Kevin Durant*. *kd* in a video game related tweet refers to *kill/death (ratio)*. Furthermore, Kevin Durant is a highly specific abbreviation – making a significance decision about what abbreviations to keep in a manually maintained list is causing some difficulty.

Rather the original simplistic plan of plainly mapping abbreviations to their manually sorted expansions as a preprocessing step, it may be a better approach to add their expansions as possible candidates before a final decision is made about which one is appropriate, hopefully with some better semantic bigram evidence. This does not fully resolve whether to keep a *kd:Kevin Durant* mapping in the list, though.

Multiple abbreviations with the same meaning

The following abbreviations are all representative of laughter: *lol, lolz, lmao, lmfao, lls, rofl*. From a semantic perspective (and for purposes of the semantic bigram index), it might be better to treat these all as the same token. Furthermore, it might be awkward to spell the full expansion out in the final result,

[3]Soundex

e.g. is it desirable to purify *LOL I'll just ignore this* to *laughing out loud I'll just ignore this*? We are considering mapping all of these to a simple *(laughter)* token.

Soundex efficacy and bigrams

Soundex was used to create the inverse phonetic index, but it may not be 100% applicable for all our needs. We noted that a large portion of Twitter slang consist of phonetic contractions, e.g. *tryna* vs *trying to.* Theoretically, we could make the appropriate adjustments by including bigrams in the inverse phonetic index so that the intended phrase *trying to* is part of the candidate correction list. However, this test case does not even work in Soundex. *tryna* maps to *t65*, while *tryingto* maps to *t6523.* We may need to make some adjustments to the existing Soundex algorithm, or find an alternate; we will investigate Daitch-Mokotoff Soundex and NYSIIS.