WORD BLENDS

Introduction

Word Blends are words formed from parts of two or more words, English “cornut” for example, is fused lexical blend of croissant and donut. Lexical blends are frequently used in Twitter application by users. This report presents a method which is used to detect potential word blends from a pre-processed list of tokens from Twitter data [4]. The method has been implemented by using Jaro similarity measure as one of the treatment steps.

Further this report encases the knowledge gained while developing this method which has been utilized to identify the true word blends.

Resources

A list of tokens from Twitter data set[4] has been used, along with a dictionary list of words for reference purpose. A true blend list has been used for confirming the results derived from the method developed for this knowledge task.

Evaluation

Method presented in this report has been evaluated using two metrics i.e. Precision and Recall.

Different treatments have been given to the raw data set to arrive at the final result and they are as follows:

a. **Preprocessing.**
b. **Candidates sufix length measurement.**
c. **Jaro similarity measurement.**
d. **Detecting potential suspects in “True Blends” list.**

Analysis

1. Fundamental

Word blends have a pattern of prefix and suffix which are parts of the two original words. Blends abridge then combine lexemes to form a new word[2].

Blends can be divided into three groups:[2]

a. **Phonemic overlap**: A syllable or part of a syllable is shared between two words.

b. **Clipping**: Two words are shortened then compounded.

c. **Phonemic overlap and clipping**: Two words are shortened to a shared syllable and then compounded.

In order to detect a potential blend candidate the fundamental concept is to detect one of the two original words using either prefix or suffix part or both.

Here in this method the suffix part of the "Twitter data"[4] tokens has been utilized and further treatments have been given to sufix in order to detect potential blend candidate.

2. <u>Different Treatments</u>

2.a. The fundamental concept of Jaro similarity depends upon the minimum number of single-character transpositions required to change one word into the other [1]. Fundamentally alone it can't be used to detect potential blend candidates as they have two different components.

2.b. A lot of the raw data can be filtered as they contain many words containing repeated characters "aaaaaa" or "hahahaha" "aaaaaaaaawwwwwwwwww" as examples. Word with length less than 3 is dropped as they can't form a blend. These steps comes as a part of preprocessing which helps to increase the precision and recall.

2.c. The suffix part of a given token is considered for identifying potential blend candidates. If the token length is upto 7, then first 2 characters is dropped and rest becomes the suffix part and if the token length is greater than 7 then first 4 characters is dropped with rest as suffix. This step provide the potential part of the second original word.

2.d. The Jaro similarity is calculated over this suffix part and the dictionary words and the most similar word is picked. The difference in length of picked dictionary word and the candidate word is calculated and if the difference is not greater than 4 then the candidate word is put in a list of potential true blend candidates. For example: {candidate word - (most similar dictionary word)<=4} → put in list of potential true blend candidates. This step is taken on the statistic observation of average length difference between a potential candidate length and the dictionary word length.

2.e. The potential true blend list of candidates are finally compared with the first column of the true blend list provided in order to obtain the final result.

An increment in the precision value has been observed after giving the treatments to the raw data and the result has been populated in the table 1 and table 2.

Steps	Precision	Recall
With "ed" "ing" & duplicates characters in tokens	0.009732	**0.933774**
Without "ed" "ing" & duplicates characters in tokens	**0.012251**	0.847682

Table 1: Precision and Recall measure

Potential blend words detected	True blends matched	Precision
10448	128	0.012251

Table 2: Detection Stats

Knowledge Gained

1. The preprocessing steps helps in reducing the potentially useless candidates which in turn results in higher precision.
2. The Jaro algorithm is a measure of characters in common, with consideration for transpositions.[3] It alone isn't much effective in detecting two words fused to form blends.But can be used to get similarity in parts of the blends.
3.The combination of preprocessing steps, the word length difference between candidate and dictionary and along with Jaro similarity results in increased precision for detecting blends in Twitter data[4].

Conclusion

Defining a true blend is complicated by the difficulty of determining which parts of a new word are "recoverable" i.e. the roots which can be distinguished [2].
But this report establishes that the process of detecting true blends in data can be made effective with increased precision by utilizing preprocessing and Jaro similarity with difference in length technique.

Reference

[1]https://medium.com/@appaloosastore/string-similarity-algorithms-compared-3f7b4d12f0ff

[2] https://en.wikipedia.org/wiki/Blend_word

[3]https://stackoverflow.com/questions/25540581/difference-between-jaro-winkler-and-levenshtein-distance

[4] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith,and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), pages 1277–1287

[5] Deri, A. and Knight, K. (2015) How to Make a Frenemy: Multitape FSTs for Portmanteau Generation. In Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 206–210

[6] Das, K. and Ghosh, S. (2017) Neuramanteau: A Neural Network Ensemble Model for Lexical Blends. In Proceedings of the 8th International Joint Conference on Natural Language Processing, pages 576–583

[7] Cook, P. and Stevenson, S. (2010) Automatically Identifying the Source Words

of Lexical Blends in English. In
Computational Linguistics, Volume 36(1)

.