Introduction

Artificial intelligence has become part of our everyday lives – Alexa and Siri, text and email autocorrect, customer service chatbots. They all use machine learning algorithms and Natural Language Processing (NLP) to process, "understand", and respond to human language, both written and spoken.

Give this NLP sentiment analyzer a spin to see how NLP automatically understands and analyzes sentiments in text (Positive, Neutral, Negative).

Although NLP and its sister study, Natural Language Understanding (NLU) are constantly growing in huge leaps and bounds with their ability to compute words and text, human language is incredibly complex, fluid, and inconsistent and presents serious challenges that NLP is yet to completely overcome.

NLP is a powerful tool with huge benefits, but there are still a number of Natural Language Processing limitations and problems:

- Contextual words and phrases and homonyms

Contextual words and phrases and homonyms

the same words and phrases can have different meanings according the context of a sentence and many words – especially in English – have the exact same pronunciation but totally different meanings.

For example:

I ran to the store because we ran out of milk.

Can I run something past you real quick?

The house is looking really run down.

These are easy for humans to understand because we read the context of the sentence and we understand all of the different definitions. And, while NLP language models may have learned all of the definitions, differentiating between them in context can present problems.

Homonyms – two or more words that are pronounced the same but have different definitions – can be problematic for question answering and speech-to-text applications because they aren't written in text form. Usage of their and there, for example, is even a common problem for humans

HOW TO SOLVE THE HOMONYMS IN TEXT FOR AI

Sentiment Classification using Word Embeddings (Word2Vec)

Sentiment Classification using pre-trained Models And Transformer

Word2Vec :

We can see that the angle between the "king" and "queen" vectors, and between the "man" and "woman" vectors, is very small, meaning that the similarity is near one. Another cool thing about these word vectors is that you can mathematically construct word analogies. So you can take the word vector for "king", and subtract the word vector for "man", that's what this dotted line represents, and then you can add the word vector for "woman" to that, that's what this dotted green line represents, and you get the exact word vector for "queen"

pre-trained Models And Transformer :

BERT is good for solving the problem of homonyms in text data because it uses a deep bidirectional transformer architecture that is capable of capturing the context and meaning of words in a sentence. It is able to do this by processing the entire sentence at once and considering the relationships between all the words in the sentence, rather than processing words in isolation.

This is particularly useful for homonyms, which are words that have multiple meanings depending on the context in which they are used. BERT is able to understand the context in which a homonym is used and use that information to determine the correct meaning of the word.

Furthermore, BERT can be fine-tuned on a specific task, such as sentiment analysis, by using a relatively small amount of labeled data. This allows the model to be adapted to a specific domain or problem, such as disambiguating homonyms in text data for sentiment analysis, with relatively little additional training data.

Overall, BERT's ability to capture the context and meaning of words in a sentence and its ability to be fine-tuned on a specific task make it well-suited for solving the problem of homonyms in text data.

Another techniques to overcome problem of Homonyms:

- Part-of-speech taggin
- Named Entity Recognition (NER)
- Word sense disambiguation
- Contextual embeddings

DATA DESCRIPTION

Overview

This is an entity-level sentiment analysis dataset of twitter. Given a message and an entity, the task is to judge the sentiment of the message about the entity. There are four classes in this dataset: Positive, Negative ,Neutral and Irrelevant . We regard messages that are not relevant to the entity (i.e. Irrelevant) as Neutral

Usage

using twitter_training.csv as the training set
and twitter_testing.csv as the test set.

Content

DATA CONTAIN 5 COLUMNS AND 74682 ROW

- INDEX : INDEX OF A ROW
- TWEET ID : ID OF TWEET
- ENTITY : TYPES OF VIDEO GAMES BRANDS
- SENTIMENT : Positive, Negative ,Neutral and Irrelevant
- TWEET CONTENT : A TWEET ITSELF

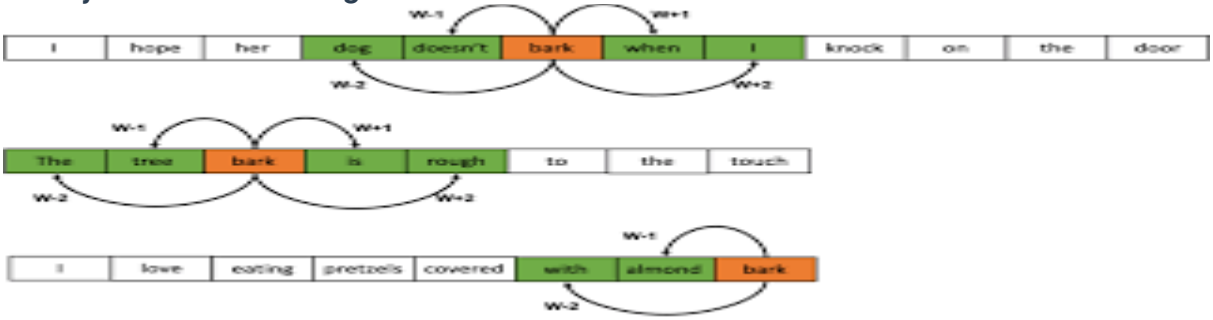
TOOLS

- Python
- Nltk
- Pre-trained models & Transformer
- Numpy
- Tensorflow

Baseline Experiments

THE GOAL → solve the problem of homonyms by Word2vec mechanism & pre-Trained model & Transformer

- Using word2vec mechanism to differentiate between meanings according the context of a sentence and many words have the exact same pronunciation but totally different meanings.



In all three sentences , you can immediately tell the meaning of bark by looking at the words around it. In practice, you would usually select a window of surrounding words and try to infer the original word's meaning by looking at the selected words. For example, in the sentences above, we can see that the meaning of bark differs from sentence to sentence because of the surrounding words known as “context words,” and we would refer to “bark” as the “center word”

- Homonyms are words that have the same spelling or pronunciation but different meanings. For example, the word "bank" can refer to a financial institution or the edge of a river. BERT addresses this problem by using a technique called contextualized word embeddings.
  Contextualized word embeddings are vectors that represent words based on their context in a sentence. BERT uses a deep neural network to analyze the entire sentence in which a word appears, as well as the surrounding sentences. This allows BERT to understand the meaning of a word based on its context, rather than relying solely on its dictionary definition.
  BERT also uses a technique called masked language modeling, which involves randomly masking out words in a sentence and then predicting the missing words based on the context. This helps BERT learn the relationships between words and their context, which allows it to better understand the meaning of homonyms.

Steps of Experiments :

1) Importing important libraires
2) Load data (Data was splited to Train & Test File)
3) Data pre-processing (drop duplicates and null rows)
4) Text pre-processing :

   Not all the information is useful in making predictions or doing classifications. Reducing the number of words will reduce the input dimension to your model. The way the language is written, it contains lot of information which is grammar specific. Thus when converting to numeric format, word specific characteristics like capitalisation, punctuations, suffixes/prefixes etc. are redundant. Cleaning the data in a way that similar words map to single word and removing the grammar relevant information from text can tremendously reduce the vocabulary. Which methods to apply and which ones to skip depends on the problem at hand.

   - Lower casing
   - Remove all the special characters - remove all single characters
   - Substituting multiple spaces with single space
   - Removing prefixed
   - word tokenize - word lemmatization
   - remove stop words
5) Word Embeddings by word2vec model
6) Bert , Roberta and vader
7) Evaluation of Trained Model on Test Set and compare between 3 different Techniques

RESULTS

- Classification Report shows the average accuracy which is 0.90. This is a good result compared to the another experiments that had done by another users of this data set . The predict function can be used on the model object to get the predicted class for the test data. Accuracy for positive and negative sentiments is better than neutral which makes sense as it is hard to distinguish the neutral comments compared to commonly used words in the positive and negative sentiment.
- Pre-trained Models have the ability to problem of Homonyms and differentiate between negative , positive and natural Tweets

CONCLUSION

Word2vec contain the potential of being very useful, and have the ability to distinguish between different meanings of word , even fundamental to many NLP tasks, not only traditional text but sometimes not efficient enough
using BERT and Transformer models can be more effective approach than Word2vec to solve the problem of homonyms. Still, their use should be considered in conjunction with other techniques and evaluated carefully based on the specific task and domain.

References

1. https://kavita-ganesan.com/comparison-between-cbow-skipgram-subword/#.YYqT32BBzIU
2. https://machinelearningmastery.com/develop-word-embeddings-python-gensim/
3. https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671
4. https://medium.com/analytics-vidhya/introduction-to-bert-and-its-application-in-sentiment-analysis-9c593e955560
5. https://www.mdpi.com/1424-8220/22/11/4157