A DISTRIBUTIONAL SEMANTICS MODEL FOR IDIOM DETECTION TMP18-17-DIST-SEMANTICS

Supervised by
Prof. Dr. Siegfried Handschuh
Dr. Jelena Mitrovic

Submitted by
Kartik Bhingaradiya - 82781
Vijay Bhuva - 86621

Contents

1 Abstract

Idiom extraction are well- known challenges with many potential applications in Natural Language Processing (NLP). Expressions, such as add fuel to the fire, can be interpreted literally or idiomatically depending on the context they occur in. We train idiom classifier on a newly gathered corpus of Wiktionary multiword definition. These gains also translate to idiom detection in sentence by simply using known word sense disambiguation algorithm to match phrases to their definition.

2 Introduction

Natural language is filled with emotion and implied intent, which are often not trivial to detect. so main challenge are idioms. Figurative language draws of off prior references and is unique to each culture and sometimes what we don't say is even more important than what we do. This, naturally, presents a Significant problem for many Natural Language Processing (NLP) applications as well as for big data analytics.

Idioms are a class of multiword expressions (MWEs) whose meaning cannot be derived from their individual constituents. Idioms often present idiosyncratic behavior such as violating selection restrictions or changing the default semantic roles of syntactic categories. Consequently, they present many challenges for Natural Language Processing (NLP) systems. For example, in Statistical Machine Translation (SMT) it has been shown that translations of sentences containing idioms receive lower scores than translations of sentences that do not contain idioms. [1]

It turns out that expressions are often ambiguous between an idiomatic and a literal interpretation, as one can see in the examples below: (A) After the last page was sent to the printer, an editor would ring a bell, walk toward the door, and holler night!" (Literal) (B) His name never fails to ring a bell among local voters. Nearly 40 years ago, Carthan was elected mayor of Tchula. . . (Idiomatic) (C) . . . that caused the reactor to literally blow its top. About 50 tons of nuclear fuel evaporated in the explosion. . . (Literal) (D) . . . He didn't pound the table, he didn't blow his top. He always kept his composure. (Idiomatic) (E) . . . coming out of the fourth turn, slid down the track, hit the inside wall and then hit the attenuator at the start of pit road. (Literal) (F) . . . job training, research and more have hit a Republican wall. (Idiomatic). [2]

3 Motivation

Several approaches have been explored in finding a better solution to this problem. However, a number of questions about automatic processing of semantic relationships specifically those that are not trivial to define and disambiguate still remain unanswered. Our Approach addresses the problem of determining automatically an idiomatic in a specific context, in our project, we only consider those expressions that are ambiguous in nature and can be interpreted either literally or figuratively depending on the context they occur in.

4 Experiments

4.1 Research

During our research to find the perfect solutions for detection of automatic idiom detection system, we have experimented following approches to detect idiom from the input corpus.

Skip-Gram Model: The skip-gram neural network model is actually surprisingly simple in its most basic form. Train a simple neural network with a single hidden layer to perform a certain task, but then we are not actually going to use that neural network for the task we trained it on! Instead, the goal is actually just to learn the weights of the hidden layer we all see that these weights are actually the **word vectors** that we are trying to learn.

Frequency Distribution:
wordfreq = [] for w in wordlist: wordfreq.append(wordlist.count(w)) print("Pairs" + str(zip(wordlist, wordfreq))) output: [('it', 4), ('age', 2)]

Nearest Neighbors Classification(Sklearn,Breute force, K-D tree, Ball tree):
The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these import pydsm.similarity as similarity ppmi.nearest$_{n}eighbors('moon', sim_{f}unc = similarity.cos$)

Cosine Similarity: The cosine similarity between two vectors (or two documents on the Vector Space) is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude.

tf-idf: Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining.

4.2 Our Approach

Our approch for automatic idiom detection is spaCy. spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. spaCy is able to compare two objects, and make a prediction of how similar they are. Predicting similarity is useful for building recommendation systems or flagging duplicates. For example, you can suggest a user content that's similar to what they're currently looking at, or label a support ticket as a duplicate if it's very similar to an already existing one. Basic functions of spaCy are Tokanization, Part-of-speech(POS) tagging, dependency parsing, Lemmatization, Sentence Boundary Detection, Similarity, Text Classification, Training,Word vectors and similarity.

How does it works?

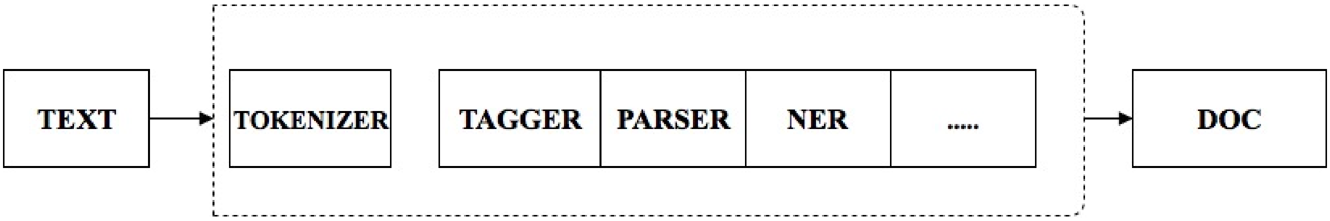


Figure 1: The flowchart of spaCy.

Training the data:

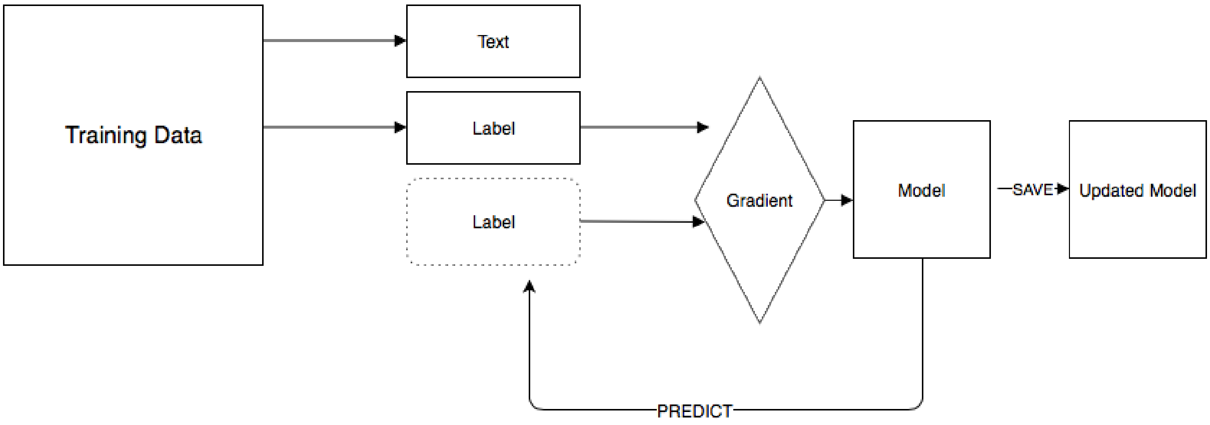
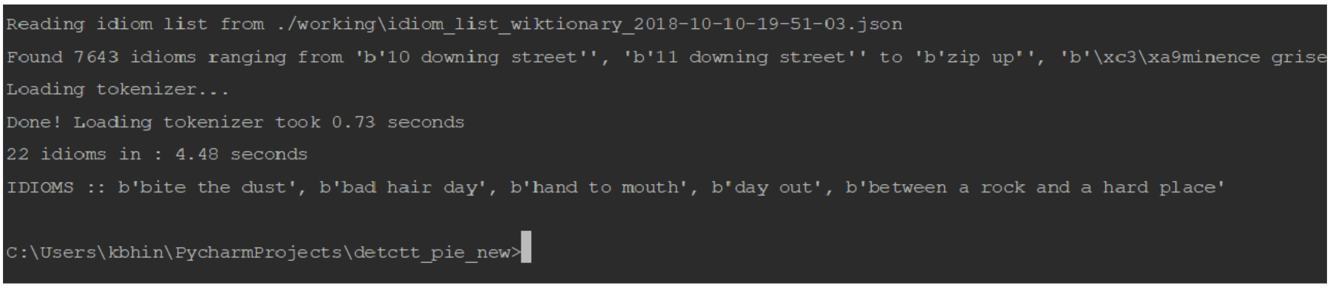


Figure 2: Train the data.

5. Results

As per our experiment, we get a result to detect idioms automatically from the dataset. We also mentioned the time that system take to identify idioms from the dataset. We also calculate the total numbers of idioms present in the data.



```
Reading idiom list from ./working\idiom_list_wiktionary_2018-10-10-19-51-03.json
Found 7643 idioms ranging from 'b'10 downing street'', 'b'11 downing street'' to 'b'zip up'', 'b'\xc3\xa9minence grise
Loading tokenizer...
Done! Loading tokenizer took 0.73 seconds
22 idioms in : 4.48 seconds
IDIOMS :: b'bite the dust', b'bad hair day', b'hand to mouth', b'day out', b'between a rock and a hard place'

C:\Users\kbhin\PycharmProjects\detctt_pie_new>
```

Figure 3: Output.

6. Conclusion

After researching the other NLP models we conclude to use spaCy library to detect automatic idioms. We have reported results, flow charts and main features of spaCy model.As a part of final model we have used Wiktionary dictionary to test and train our model, apart from that we have also tested BNC data set.

6.1 Software Requirements

- Beautifulsoup 4.5.1
- spacy 1.9.0
- lxml 4.2.1
- Python 3.6
- NLTK 3.2.1
- Stanford Core NLP Parser

7. References

[1] Caroline sporleder, linlin li, philip john gorinski, xaver koch Idioms in Context: The IDIX Corpus.

[2] https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/.

[3] A Distributional Semantics Model for Idiom Detection: The Case of English and Russian.

[4] Automatic Idiom Recognition with Word Embeddings. jing peng(b) and anna feldman.

[5] ICE: Idiom and Collocation Extractor for Research and Education. Vasanthi Vuppuluri Shahryar Baki, An Nguyen, Rakesh Verma.