Report

Anna Liednikova

January 11, 2019

Contents

1 Introduction

This project was set within the framework of a collaboration with the ALIAE startup where the aim is to develop a chatbot to collect information from clinical patients. The main idea is to replace strictly defined surveys by a more natural conversation in order to let users express themselves freely so that more information could be collected.

ADD: The chatbot consists of three main modules: NLU (Natural Language Understanding: interpreting the user input), Dialog Managment (Deciding on how to respond to the user input) and NLG (Generating the system response).

ADD: In this project, we focus on the NLU step which consists in analysing the user utterance (does the user speaks about pain, physical activity etc. ?) and detecting at least one most probable intent of user message so that the chatbot can choose the right strategy for fulfilling information.

ADD: ADD an example showing a user input and the expected representation ie intent + entity

A small manually annotated dataset (roughly 100 sentences for each category) was available.

ADD: say that this dataset is not enough for learning a good NLU models. Hence your work focuses on exploring automatic ways of extending the training data and learning NLU models from these extended data.

ADD: Explain report structure: which section focuses on what?

REMOVE: We have tried models for sentence representation by word and the whole context in order to use semi-supervised learning by gradually increasing train data by labelling external sources.

2 Literature review

[?] MODIFIED: present an approach for automatically generating additional training data for event extraction systems. First, they trained a baseline classifier on available data. Then, they cluster external data to obtain cluster of paraphrases using the NewsSpike method introduced by [?]. They then label the clusters using the baseline model trained on the initial dataset of labelled data. Combining the new labelled data and the original one, they then retrained the event extractor.

3 Methodology

ADD: We follow [?] methodology. Instead of using [?]'s methods for identifying clusters of related sentences however, we explore different ways of representing sentences using deep learning approaches. We then apply clustering to the resulting sentence representations. Finally, we investigate the impact of the extended labeled data on classification.

3.1 Building Sentence Representations

We create sentence representations in two steps using machine and deep learning techniques. First, we map words to continuous representations. Second, we combine these word representations into sentence representations.

3.1.1 Word Representations

We explore two ways of building word representations: Word2vec and LDA.

ADD: You need to explain what LDA and Word2Vec are and how they represent word/sentences

REMOVE: The most popular word-basis techniques to represent sentences are LDA (Latent Dirichlet Allocation) and Word2Vector model. We used their implementation from gensim library in order to build models based on the initial labelled dataset.

3.1.2 Sentence Representations

MODIFIED: We construct sentence representations out of the word representations described in the previous section using two types of neural networks: PositionwiseFeedForward [?] and BiLSTM ADD: bibref.

ADD: Here you need to explain the theory behind it. Explain how LSTM and the PositionwiseFeedForward build a sentence representation out of word embeddings.

Bi-LSTM Representations.

Tranformer Representations. ¡¡¡¡¡¡ HEAD ======= The most popular word-basis techniques to represent sentences are LDA (Latent Dirichlet Allocation) and Word2Vector models. We used their implementation from gensim library in order to build models based on the initial labelled dataset.

Preprocessed tokenized sentences were used to create dictionary and were passed to W2V model as it is and in form of Bag-Of-Word to LDA model.

For selecting the best model we changed space dimension (topics for LDA and features for W2W). Preferred number of features for W2V model is 200. It's the start point for our model.

The good point of using this models is that they can be used for both word and sentence representation since one word can be treated as a small sentence. ¿¿¿¿¿¿¿ 1fd91297f0d748d6268bc796f027ba6b4c129a5a

3.2 Clustering Sentences

MODIFIED: Using the sentence representations described in the previous section, we apply clustering to group together sentences that are similar. We compare three clustering algorithms:

ADD: Briefly explain the key feature of each clustering algorithm

- K-Means (KM)
- AgglomerativeClustering(ward) (AG)
- GaussianMixture (GM)

¡¡¡¡¡¡¡ HEAD We set the number of clusters to the number of intents (20) which allows using not only purity and Silhouette coefficients to evaluate the clusters but also homogeneity and completeness to get the idea how resulting groups correlate with initial intents. Also, for each model, we plot a confusion matrix with a background gradient to visualize how well clusters separate different intents.

3.3 Classifying Sentences

Train on labelled data ======= The number of clusters was set to the number of intents (20) that allows using not only purity and Silhouette coefficients to evaluate them but also homogeneity and completeness to get the idea how resulting groups correlate with initial intents. Also, for each model, we plotted a confusion matrix with a background gradient to visualize how well clusters separate different intents. ¿¿¿¿¿¿¿ 1fd91297f0d748d6268bc796f027ba6b4c129a5a

Test on unlabelled data

4 Datasets

4.1 Labelled Data

The initial dataset is created manually covering 20 main possible users intents. Each sentence represents one intent. For, example,

(1) **Text:** After sleep, for 2-3 hours, I am better and then start feeling tired again
Intent: sleep

The distribution of labels is shown in Figure **??**. There are in average, 3.4 words per sentence (min: 0, max: 12, std: 2.16367). The total number of sentences is 3305 and the vocabulary consists of 1882 content words (after removing stopwords).

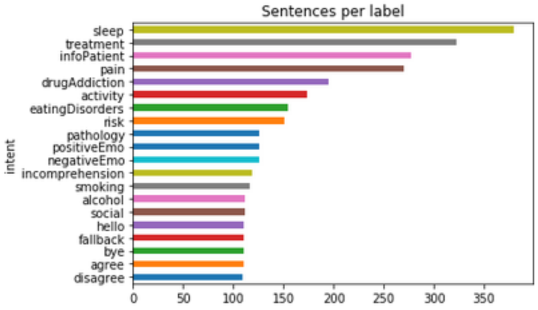

Figure 1: Label distribution in dataset

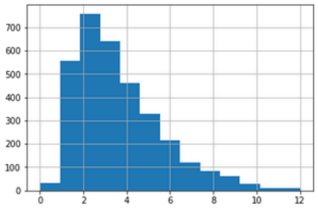

Figure 2: Sentence dist

4.2 Forum data

To create additional training data, we extracted textual data from HealthBoards ADD: URL, a medical forum web portal that allows patients to discuss their ailments.

We scraped 272,553 unique posts contained in each category. The post were then segmented into sentences, tokenized and lemmatized using NLP libraries. Stop words were removed.

We compared two libraries, NLTK and SpaCy.

For example, in word tokenization they give different results that can influence not only simple statistics but meaning too. Some example of different tokenization can be seen in table **??**. Though concating words to ones like ’flulike’ or ’35mg’ or ’longterm’ sometimes gives more robust and concrete meaning in case of big dataset, in our case it seems better to stay with spacy way of tokenization in order to have smaller and more simple vocabualary.

For stopwords removing there were three options: nltk, spacy and the longest one. The last option was rejected due to containing words like ’want’, ’stop’, ’successfully’ etc. that can be useful for detecting basic intents like positive or negative emotion, social. Finally nltk one was selecting because of containing shorts like ’m’ from ’am’, ’ve’ from ’have’. Final dictionary contained 1882 words. Also all numbers were changed to num. Chart (**??**) looks fine.

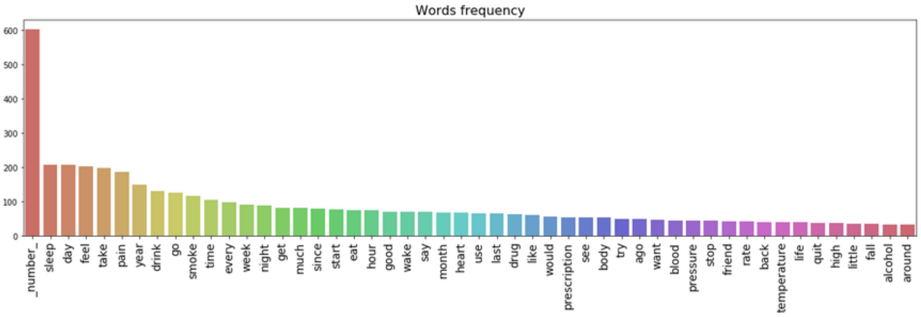

Figure 3: Words frequency

Length ¡ 30.

Next problem is empty sentences. But they don’t change the dataset much.

intent fallback 12 disagree 11 hello 4 agree 3 incomprehension 2 positiveEmo 1

array([’what?’, ’same that again’, ’I am up’, ’same’, ’can you’, ’do that’, ’do this’, ’Will do’, ’no’, ’no i will not’, ”No i don’t”, ’no’, ’no i will not’, ”No i don’t”, ”What’s up?”, ”What’s up?”, ’he’, ’a’, ’d’, ’i’, ’m’, ’o’, ’s’, ’t’, ’y’,

NLTK	SpaCy
[’i’, ’wouldnt’, ’go’, ’to’, ’sleep’, ’until’, ’like’, ’5’, ’6’, ’or’, ’8am’]	[’i’, ’would’, ’nt’, ’go’, ’to’, ’sleep’, ’until’, ’like’, ’5’, ’6’, ’or’, ’8’, ’am’]
[’that’, ’is’, ’totally’, ’wrongheaded’]	[’that’, ’is’, ’totally’, ’wrong’, ’headed’]
[’i’, ’am’, ’in’, ’the’, ’process’, ’of’, ’tapering’, ’from’, ’suboxone’, ’longterm’, ’use’]	[’i’, ’am’, ’in’, ’the’, ’process’, ’of’, ’tapering’, ’from’, ’suboxone’, ’long’, ’term’, ’use’]
[’i’, ’had’, ’an’, ’onandoff’, ’opiateopioid’, ’habit’, ’from’, ’about’, ’2010’]	[’i’, ’had’, ’an’, ’on’, ’and’, ’off’, ’opiate’, ’opioid’, ’habit’, ’from’, ’about’, ’2010’]
[’i’, ’have’, ’flulike’, ’pathologysymptom’]	[’i’, ’have’, ’flu’, ’like’, ’pathologysymptom’]
[’i’, ’have’, ’exerciseinduced’, ’insomnia’]	[’i’, ’have’, ’exercise’, ’induced’, ’insomnia’]
[’i’, ’m’, ’supposed’, ’to’, ’take’, ’6’, ’35mg’, ’tablets’, ’a’, ’day’, ’but’, ’i’, ’have’, ’taken’, ’20’, ’today’]	[’i’, ’m’, ’supposed’, ’to’, ’take’, ’6’, ’35’, ’mg’, ’tablets’, ’a’, ’day’, ’but’, ’i’, ’have’, ’taken’, ’20’, ’today’]

Table 1: Tokenization comparison

	# Sentence	Avg Stce Size (min/max)	Vocab. Size
Unlabelled Data	3305	3.4 (0/12)	1882
Labelled Data		10.93 (0/2027)	

Table 2: Labelled and Unlabelled Data

’I will’, ’Will do’, ’No’, ’no’, ’No it is not’, ”No I don’t”, ’No’, ”I’m here”], dtype=object)

Finally, the corpus consists of N sentences. Table **??** and Figure **??** show the dataset statistics.

’min’: 0, ’max’: 2027, ’mean’: 10.932588521491454, ’std’: 8.964716092053479 ’min’: 3.2580388329566468, ’max’: 35.23600634007455, ’mean’: 11.224462266261876, ’std’: 6.963225985041148

Data should be divided in subsets by increasing sentence length because of the difference in mean values for both datasets.

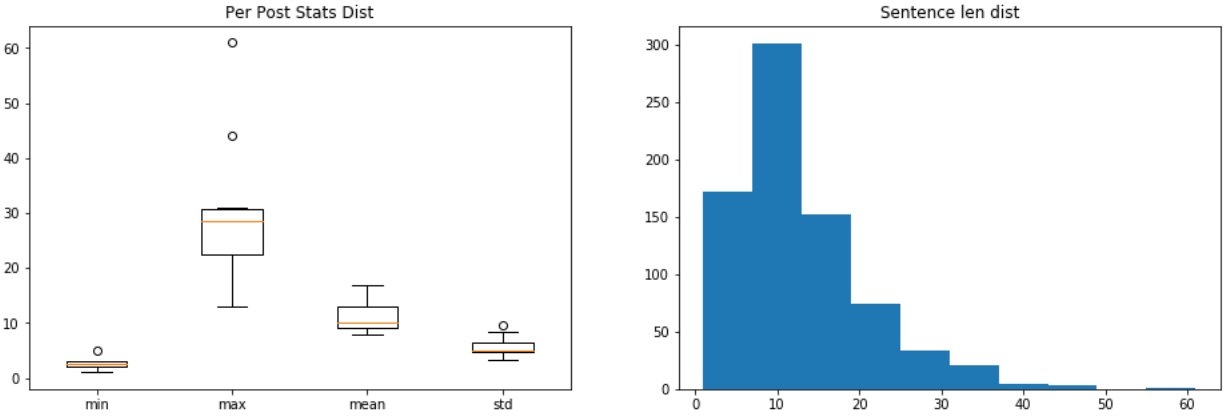

Figure 4: Text stat

5 Experiment

5.1 Clustering

5.1.1 Evaluation metrics

5.1.2 W2V model

Comparison table

WE	Cluster	purity	Silhouette	homogeneity	complete	Clf CV score
Defenders	KM AC GM					
M	KM AC GM					
Forward	FW					
S	KM AC GM					

5.1.3 LDA model

5.1.4 GloVe pretrained

5.1.5 Google News W2V pretrained

5.1.6 CNN by word

From simple encoder, w2v, lda, w2v + lda

5.1.7 BiLSTM by word

From simple encoder, w2v, lda, w2v + lda

5.1.8 Overall comparison

5.2 Classifier

5.2.1 Evaluation metrics

5.2.2 Results for each model

6 Conclusion

7 References