Data expansion for semantic classification

Anna Liednikova, Claire Gardent

January 18, 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.

The chatbot consists of three main modules:

- NLU (Natura Language Understanding): interpreting the user input
- Dialog Managment: deciding on how to respond to the user input
- NLG: generating the system response

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 later 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. But that's not enough for learning a good NLU model and meet the variety of users possible inputs. Annotating sentence takes plenty of time, so some automated or semi-automated way should be chosen. So this project 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?

First, we do the literature review of approaches applied to similar problems. Later in the methodology part, we will cover four parts: building sentence representation model, evaluate it by clusterisation, train a classifier for labelling the data and application semi-supervised learning for extending the dataset. Next two our datasets and preprocessing routine will be described. Experiment part contains a quantitative and qualitative description of the models and final results. In the end, we will sum up the work have been done and give some assumption for future improvements.

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

MODIFIED: Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is charac- terized by a distribution over words.

LDA assumes the following generative process for each document **w** in a corpus **D**:

- 1.Choose N $\sim$ Poisson(ξ).
- 2. Choose $\theta \sim$ Dir(α).
- 3. For each of the N words w_n :
- (a) Choose a topic $z_n \sim$ Multinomial(θ).

- (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

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.

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.

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:

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

ADD: Briefly explain the key feature of each clustering algorithm

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

After choosing the best model for word embedding we train classifier to label data with intents. The most common approach in similar task is to use SVC. So, we create balanced train and test dataset and validate model by cross validation score.

Another classical approach for classification is Random Forest Classifier.

3.4 Semi-supervised approach

First, we find the best clusterization for the forum data, later we classify each item and omit ones with low probability of elements at threshold Theta. In result some clusters will be empty from the beginning and some of them will be ommited due to small population.

Elements of clusters that are left will be labelled by the majority. Usually, they are pure. Later we select N labelled items for each category and extend train data with them to retrain word embedding and classificatin models.

We repeat this cycle until we won't be able to extend train dataset by additional samples or classifier score will stop to grow.

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,

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 1. 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).

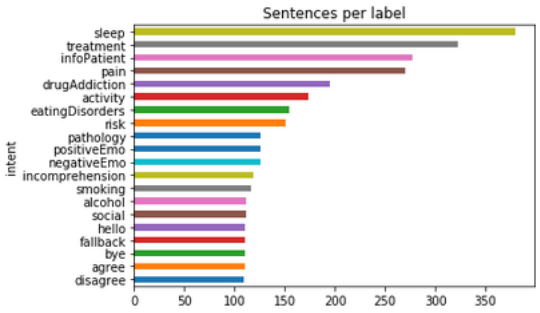


Figure 1: Label distribution in dataset

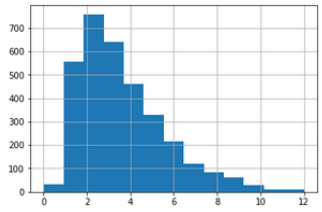


Figure 2: Sentence dist

Total number of sentences is 3305. Found 1882 words after filtering through stopwords list.

4.2 Preprocessing routine

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 1. 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 (3) looks fine.

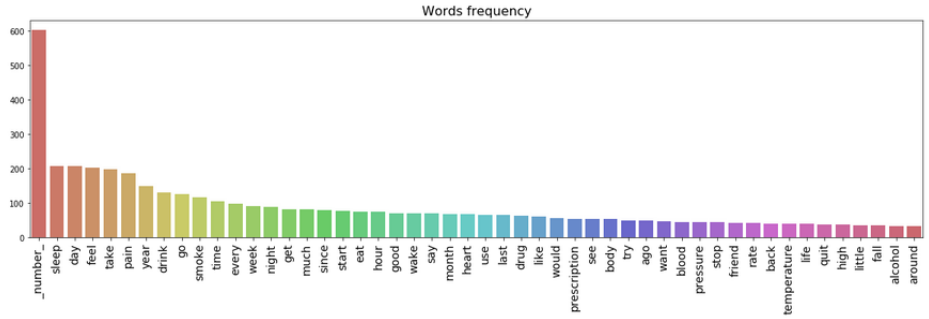


Figure 3: Words frequency

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

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

4.3 Forum data

To improve initial classifier we needed to extend dataset by much more data and more naturally constracted one. No test set is available was available at the beginning of the project. To solve this problem, we opted to created our own test set following Zhang et al (2015) and Sondhi et al (2010) example.

HealthBoards is a medical forum web portal that allows patients to discuss their ailments. We scraped 272553 unique posts contained in each of 238 categories. Finally, the corpus consists of N sentences. Table 2 and Figure 4 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

	# 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

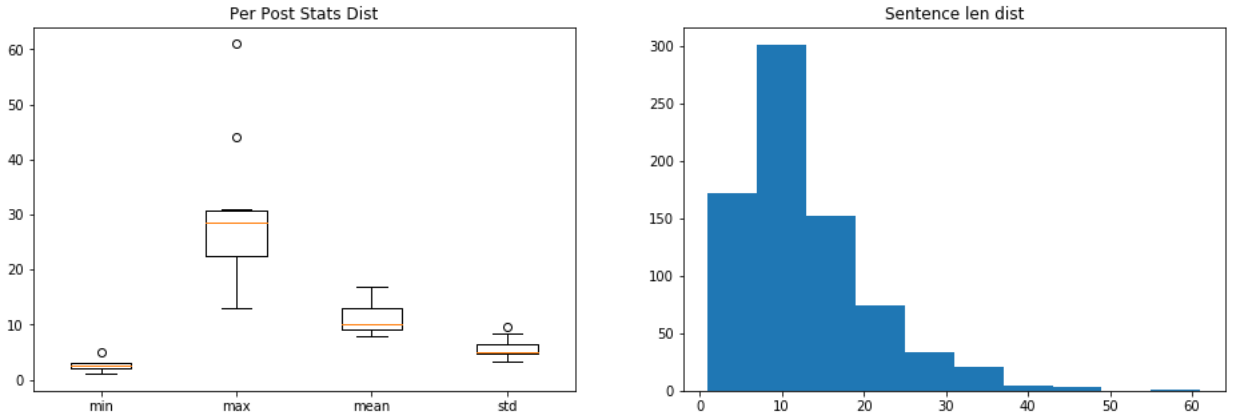


Figure 4: Text stat

Because our intents are not that precise as forum ones we should select particular boards with relevant information to be able gradually expand their context. Result is in table 3 We should start with the data that is most similar to the initial one, so we choose categories of boards that are similar to our intents.

Category of board	# of posts
digestive-disorders	5064
addiction-recovery	3644
sleep-disorders	1748
smoking-cessation	937
eating-disorder-recovery	762
chronic-pain	735
chronic-fatigue	662
stress	415
family-friends-addicts-alcoholics	312
pain-management	25

Table 3: Selected categories

Also, data should be divided in subsets by increasing sentence length because of the difference in mean values for both datasets. So after tokenizing posts into sentences we calculate it's length. For the each iteration we should leave sentences with mean+std words in cleaned text.

5 Experiment

5.1 Clustering

5.1.1 Evoluation metrics

For every model following metrics and their average were calculated: purity score, Silhouette Coefficient, homogeneity and completeness scores. Later for each parameter average among clustering models calculated for each score in order to get both table and plot.

Also for each model confusion matrix created between cluster labels and initial intents.

5.1.2 W2V model

Comparison table

features num	purity	silhouette	homogeneity	completeness
10	0.183157	0.0680834	0.108901	0.12661
20	0.182854	0.0506986	0.111099	0.124805
30	**0.185477**	0.0683824	0.110799	0.132563
50	0.184569	0.0647393	**0.111876**	0.131964
100	0.182148	**0.0813837**	0.108022	0.138329
150	0.183661	0.0298524	0.106401	0.141179
200	0.176803	0.0581627	0.103708	0.141973
300	0.176097	0.0432732	0.106379	0.151118
400	0.169642	0.0626822	0.100455	0.148768
500	0.168533	0.0653733	0.0976706	0.146799
550	0.17176	0.0584935	0.0994422	**0.151185**
600	0.169138	0.0574187	0.0980455	0.150695

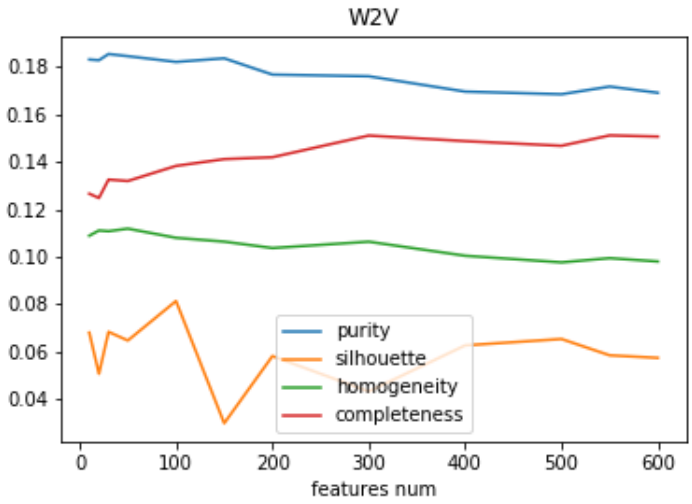


Figure 5: LDA scores

Later we will try both 10 and 100 features.

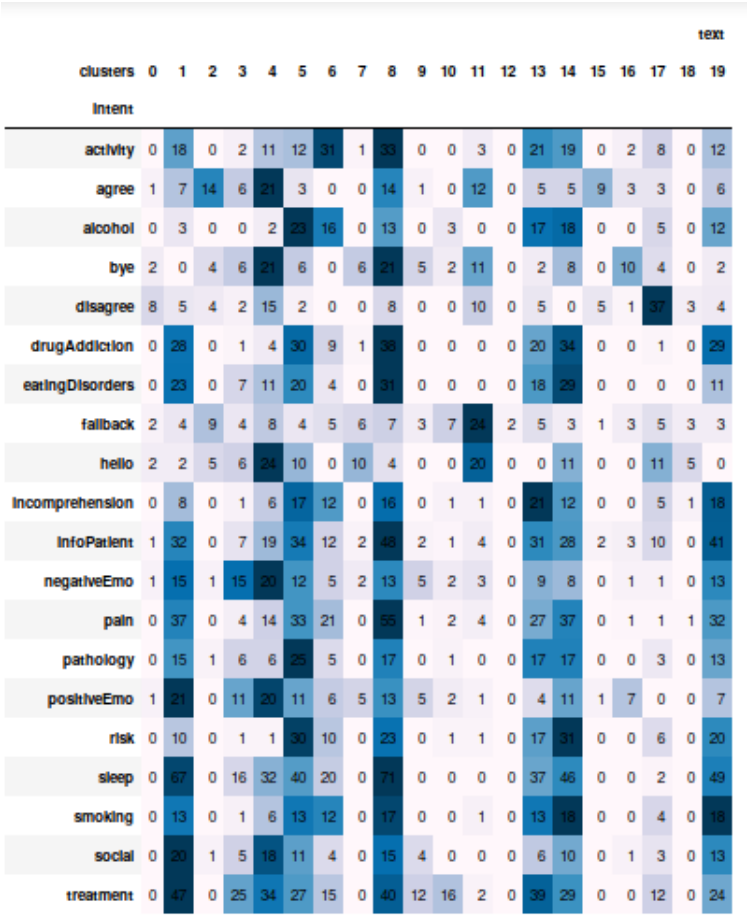


Figure 6: W2V confusion matrix

5.1.3 LDA model

num	purity	silhouette	homogeneity	completeness
10	0.253656	0.461589	0.157498	0.178928
20	0.259203	0.378276	0.168191	0.182203
30	0.251538	0.331069	0.157755	0.168448
50	0.240545	0.287032	0.147430	0.166824
100	0.266465	0.187596	0.168647	0.189221
150	0.256077	0.167888	0.161589	0.185054
200	0.263843	0.149212	0.175117	**0.194913**
300	0.250025	0.151526	0.165607	0.184314
400	**0.272718**	0.192777	**0.177547**	0.192270
500	0.259506	**0.203725**	0.174925	0.192831
550	0.229450	0.16620013	0.149270	0.158548
600	0.254261	0.191729	0.171744	0.186001

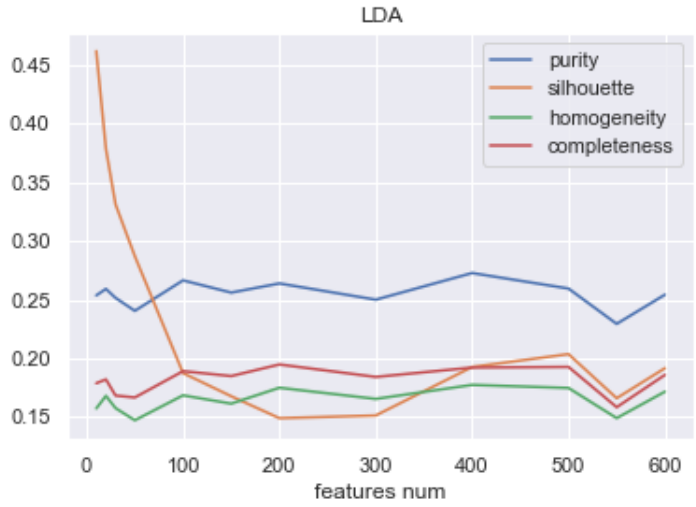
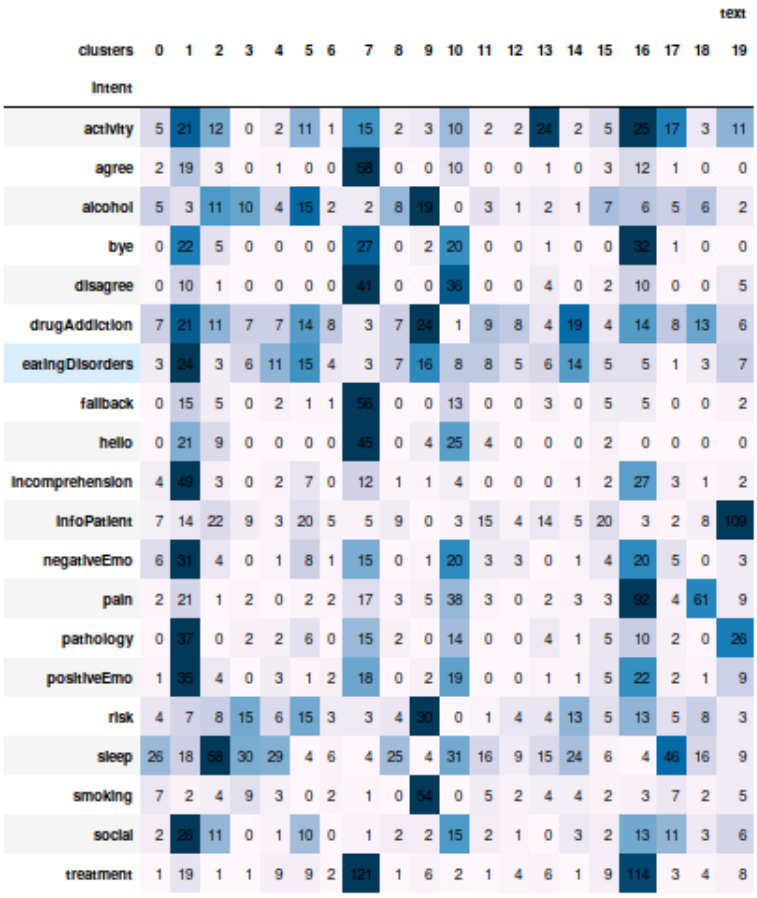


Figure 7: LDA scores

The best model - is LDA with 400 topics according to scores and with priority to purity and homogeneity ones.

Thanks to confusion matrix that is constructed based on cluster and intent labels we can make assumpsions about possible errors and difficulties and figure out what connections our model distinguish well. Though different clustering algorithms still captures different connection between intents, there are some common groups for all of them: fallback and hello; alcohol, drugAddiction and risk; social and emotions.



text																				
clusters	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Intent																				
activity	5	21	12	0	2	11	1	15	2	3	10	2	2	24	2	5	25	17	3	11
agree	2	19	3	0	1	0	0	58	0	0	10	0	0	1	0	3	12	1	0	0
alcohol	5	3	11	10	4	15	2	2	8	19	0	3	1	2	1	7	6	5	6	2
bye	0	22	5	0	0	0	0	27	0	2	20	0	0	1	0	0	32	1	0	0
disagree	0	10	1	0	0	0	0	41	0	0	36	0	0	4	0	2	10	0	0	5
drugAddiction	7	21	11	7	7	14	8	3	7	24	1	9	8	4	19	4	14	8	13	6
eatingDisorders	3	24	3	6	11	15	4	3	7	16	8	8	5	6	14	5	5	1	3	7
fallback	0	15	5	0	2	1	1	56	0	0	13	0	0	3	0	5	5	0	0	2
hello	0	21	9	0	0	0	0	45	0	4	25	4	0	0	0	2	0	0	0	0
incomprehension	4	49	3	0	2	7	0	12	1	1	4	0	0	0	1	2	27	3	1	2
infoPatient	7	14	22	9	3	20	5	5	9	0	3	15	4	14	5	20	3	2	8	109
negativeEmo	6	31	4	0	1	8	1	15	0	1	20	3	3	0	1	4	20	5	0	3
pain	2	21	1	2	0	2	2	17	3	5	38	3	0	2	3	3	92	4	61	9
pathology	0	37	0	2	2	6	0	15	2	0	14	0	0	4	1	5	10	2	0	26
positiveEmo	1	35	4	0	3	1	2	18	0	2	19	0	0	1	1	5	22	2	1	9
risk	4	7	8	15	6	15	3	3	4	30	0	1	4	4	13	5	13	5	8	3
sleep	26	18	58	30	29	4	6	4	25	4	31	16	9	15	24	6	4	46	16	9
smoking	7	2	4	9	3	0	2	1	0	54	0	5	2	4	4	2	3	7	2	5
social	2	26	11	0	1	10	0	1	2	2	15	2	1	0	3	2	13	11	3	6
treatment	1	19	1	1	9	9	2	121	1	6	2	1	4	6	1	9	114	3	4	8

Figure 8: LDA confusion matrix for gaussian mixture

5.1.4 GloVe pretrained

5.1.5 Google News W2V pretrained

We used word2vec google-news model with 300 dimensional feature space.
CV 0.418 + 0.057

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

model	features	purity	silhouette	homogeneity	completeness
W2V	10	0.192032	0.077404	0.108761	0.127672
W2V	100	0.184266	0.065381	0.109944	0.144074
LDA	400	0.272718	0.192777	0.177547	0.192270

5.2 Classifier

5.2.1 Evoluation metrics

5.2.2 Results for each model

5.3 Semi-supervised learning

6 Conclusion

7 References