# Data expansion for semantic classification

Anna Liednikova
Supervisor: Claire Gardent

February 4, 2019

## Contents

# 1 Introduction

This project was set within the framework of a collaboration with the ALIAE startup whose 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 two main modules:

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

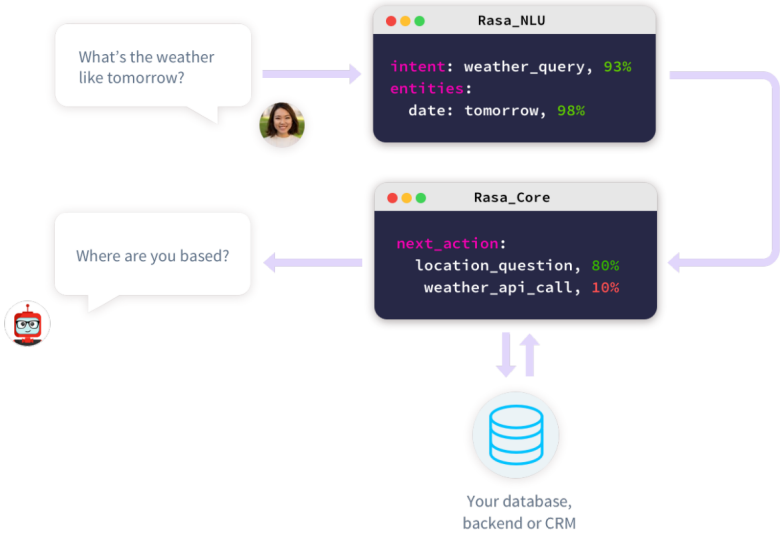


Figure 1:

In this project, we focus on natural language understanding and consider a simplified notion of understanding where each user input is mapped to an "intent". Given a pre-defined and finite set of intents (Does the user speaks about pain, physical activity etc. ?), the aim of the NLU module is to assign

each user input an intent[1]. The detected intent is then used, in conjunction with additional information extracted from the previous dialog interactions, to determine how to respond (cf. Figure 1[2]).

## 2 Goals and Methodology

The NLU module is a classifier which given a user input, predicts the corresponding intent. Our goal is to improve this classifier starting from a small training set (roughly 100 utterance/intent pair for each intent) and automatically expanding this labelled data. To this end, we developed a semi-supervised data expansion technique which permits expanding the initial data with utterances extracted from a medical forum. We show that the expanded data permits improving the classifier by ADD: ADD NB of improvment points accuracty points.

This report is structured as follows. Section 3 briefly reviews the state of the art and situates our proposal with respect to related work. Section 4 presents the methodology used for data expansion. Section 5 describes the datasets used. Section 6 reports on the experimental setting and on the results. Section 7 concludes with pointers for further research.

## 3 Literature review

To use supervised text classification methods, it is important to have a dataset of a reasonable size in order to make their training efficient. When only a small set of labelled documents is available, the semi-supervised learning becomes a way-out and helps to resolve tasks by comparing the distributions between labelled and unlabelled specimens.

[**?**] presented an approach for automatically generating additional training data for event extraction systems. First, they trained a baseline classifier on available gold data. Then, they split external data to obtain clusters 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

[1] We make the simplifying (and incorrect) assuption that the user input contains a single intent.

[2] The figure actually shows a more complex notion of understanding where the user utterance is mapped not only to an intent but also to a set of entities. In this work, we focus on how to improve intent detection and leave entity detection for future research.

then retrained the event extractor and achieved significant improvement of F1 score.

There are many models that encode sentences into fixed-size representations. LDA and Word2Vec models are the most popular word embedding model which show their effectiveness on the continuation of a long time and a wide range of tasks[?] [?]. [?] also showed the efficiency of BiLSTM-max model for capturing word context in encoding sentences.

ADD: mention about pretrained embeddings and svc [?]

SVM is considered to be a hands-on approach in textual classification [?]. These classifiers are fast, reliable, work well in both linear and non-linear separation and can effectively deal with sparse data, and have a great ability to generalize knowledge in multidimensional spaces [?]. Last two advantages especially important due to the common use of high dimension in sentence representation task and its sparse nature in case of LDA.

# 4 Methodology

Our methodology for data expansion falls into two main steps. First, we compare different methods for representing utterances using both clustering and classification (Sections 4.1 and 4.2). Second, based on the best sentence representations, we gradually increment the size of the labelled data and iteratively train new classifiers on the extended data (Section 4.3).

## 4.1 Building Sentence Representations

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

**LDA.** One of the most common methods for constructing thematic models is the Latent Dirichlet Allocation (LDA) [?] that models a document as a distribution of topics and a topic as a distribution of words. Here a document is a sentence. Hence a sentence is represented as a distribution of topics.

Let's consider a toy LDA model produces following topics:

```
[(0,
  '0.089*"_num_" + 0.025*"day" + 0.022*"pain" + 0.020*"take" +
   0.018*"year" + 0.017*"feel" + 0.015*"smoke" + 0.014*"sleep"'),
 (1,
  '0.024*"good" + 0.022*"get" + 0.021*"feel" + 0.021*"blood" +
   0.021*"want" + 0.020*"pressure" + 0.014*"understand" '),
```

```
(2,
 '0.067*"go" + 0.048*"sleep" + 0.023*"_num_" + 0.021*"bed" +
  0.018*"fall" + 0.018*"hour" + 0.017*"wake" + 0.016*"asleep" '),
(3,
 '0.040*"drink" + 0.030*"life" + 0.025*"much" + 0.020*"eat" +
  0.017*"end" + 0.016*"go" + 0.013*"time" + 0.012*"body" ')]
```

And here is an example representation for a sentence/

```
'I can go for months and at least a year or so without drinking',
[0.61565    , 0.03821149, 0.32365379, 0.02248472]
```

If we want represent just one word we can just treat it like a small sentence and have the same output.

```
'sleep',
[(0, 0.21281178), (1, 0.13255174), (2, 0.5765724), (3, 0.07806411)]
```

The LDA algorithm is based on the Dirichlet prior distribution and assumes a bag-of-words model - a model for analyzing texts that takes into account only the frequency of words, but not their order. This model is well suited for thematic modeling, since it permits detecting implicit relationships between words. The LDA method performs soft clustering and assumes that each word in the sentence is generated by some latent topic, which is determined by a probability distribution on the set of all words of the text.

**Word2Vec.** Word2Vec representations are low dimensional vectors also called embeddings which are built based on textual cooccurences and learned using shallow, two-layer neural networks [**?**]. Given a large corpus of text as input, Word2Vec produces a vector space where words that share common contexts in the input corpus will be mapped to vectors that are close to one another in the vector space.

To build sentence representations, we simply average the Word2Vec vectors of the content words present in that sentence.

**Bi-LSTM.** The general idea of BiLSTM layer is repeating the first recurrent layer in the network so that they are next to each other and then providing the straigh input sequence as input to the first level and an inverse copy of it for the second.

In more formal way, for the sequence of words $T\{w_t\}_{t=1,...,T}$ bidirectional LSTM computes a set of T vectors $\{h_t\}_t$. For $t[1,.,,,T]$, $h_t$ is the combina-

tion of forward and backward LSTM, that read sentences in two opposite directions.

We experiment with two ways of combining a changing number $(h_1, ..., h_T)$ to form a vector of a fixed size, either by choosing the last hidden state $h_T$ (BiLSTM-last), or either by selecting the maximum value over each dimension of the hidden units (max pooling) (Collobert and Weston, 2008) or by considering the average of the representations (mean pooling). The choice of these models is motivated by their demonstrated effectiveness in embedding models (Conneau et al., 2018)

### 4.2 Clustering Sentences

Using the sentence representations described in the previous section, we apply clustering to group together sentences that are similar. We set the number of clusters to the number of intents (20) and we use not only purity and Silhouette coefficients to evaluate the clusters but also homogeneity and completeness to better assess how the resulting clusters 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. Based on these metrics and vizualisation, we determine which method produces representations that best support intent detection.

We compare three clustering algorithms:

- K-Means (KM)
- Agglomerative Clustering (AG)
- Gaussian Mixture (GM)

**K-Means.** The basic idea is to define K centroides one for each cluster. It is best to place them as far apart as possible. The next step will be to accept each point belonging to this data set and associate it with the precision of the centroid. If no point is under consideration, the first stage is completed. At the moment, we need to re-calculate K's new centroids, as the cluster barycenter that stems from the previous step. After these new centroids appear on us, a new binding must be made between the same points of the data set and the nearest new center of gravity. A loop is created. As a result of this cycle, we can see that k centroids change their location step by step until any changes are made. In other words, centroids are no longer moving.

The algorithm is aimed at minimizing the target function, in this case, the quadratic error function. Target function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} (\|x_i^{(j)} - c_j\|)^2 \tag{1}$$

where $(\|x_i^{(j)} - c_j\|)^2$ is the chosen distance between the data point $x_i^{(j)}$ and center $c_j$, it is an indicator of the distance n of the data points from their respective cluster centers.

Centroid is defined as follows:

$$c_j = \frac{1}{N} \sum_{i=1}^{N} x_i^{(j)} \tag{2}$$

where $x_i^{(j)}$ is the point belonging to the cluster $J$, $N$ - total number of points in the cluster.

**Agglomerative Clustering** is a kind of hierarchical clustering with 'bottom-up' approach when new clusters are created by combining smaller clusters and, thus, a tree is created from leaves to trunk [**?**]. The main feature of the agglomerative hierarchical clustering method is that if we want to get a different number of clusters, we will not need to restart the algorithm, because the whole tree is calculated, and then we say that we want to cut it in some place.

Different linkage criterions can be used to calculate distance between sets of observation. Wards method minimizes the variance of the clusters being merged. This method is used for tasks with closely spaced clusters. The euclidian distance between clusters is taken as the increment of the sum of squares of the distances of objects to the center of the cluster, obtained as a result of their combination:

$$\Delta = \sum_{i} (x_i - \bar{x})^2 - \sum_{x_i \in A} (x_i - \bar{a})^2 - \sum_{x_i \in B} (x_i - \bar{b})^2 \tag{3}$$

At each step of the algorithm, these two clusters are combined, which lead to a minimal increase in variance.

**Gaussian Mixture.** In statistics, a mixed model is a probabilistic model for representing the presence of subpopulations in the general population

in accordance with the mixture of distributions, without requiring that a separate observation belong to a specific subgroup.

A Gaussian mixture model [**?**] represents a weighted sum of M Gaussian densities elements for D-dimensional continuous-valued data vectors:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \sum_i) \tag{4}$$

where each Gaussian densities element defined as

$$g(x|\mu_i, \sum_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sum_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)' \sum_i (x-\mu_i)} \tag{5}$$

with mean vector $\mu_i$ and covariance matrix $\sum_i$. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$.

GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

### 4.3 Semi-Supervised approach to Data Expansion and Classification

After choosing the best model for representing sentences, we iteratively expand the training data and train the corresponding classifiers as follows.

- **Step 1: Classification.** Apply the classifier trained on the currently available labelled data to a set of unlabelled sentences;
- **Step 2: Clustering.** Clusterize these unlabelled sentences into 200 clusters
- **Step 3: Filtering.** Use the probability and the label assigned by the classifier to each unlabelled sentence to :
  - Eliminate sentences whose probability is below a given threshold
  - Eliminate clusters with less than $m$ members
- **Step 4: Data expansion.** Add to training data the $n$ best labelled items for each intent.
- **Step 5: Classifier Training.** Train a new classifier using both the previous and the new labelled data.

**Classification.** For classification, we use SVC (Support Vector Classifier). The main idea of the method is transfering the initial vectors into a space of higher dimension and searching for a separating hyperplane with the maximum gap in this space. Two parallel hyperplanes are built on both sides of the hyperplane separating the classes. The separating hyperplane will be the one that maximizes the distance to two parallel hyperplanes. The algorithm works under the assumption that the greater the difference or the distance between these parallel hyperplanes, the less will be the average classifier error. Due to the small size for dataset it's essential to create balanced training and test datasets.

The initial labelled dataset was provided by ALIAE (gold data). We split our gold dataset into train and test sets and systematically use the test set for evaluation and comparison of the classifiers learned on the different datasets produced during data expansion.

At each iteration during data expansion, we use the classifier trained on the data created at the previous iteration to label unlabelled sentences extracted from some external source.

**Clustering.** To expand the training data, we retrieve utterances from a medical forum. The unlabelled data is then built based on these utterances after some filtering is applied to eliminate sentences that are either out of topic or too complex (cf. Section 1). Next we build representations for these unlabelled sentences using the techniques described in Section 1 and we apply clustering to create 200 clusters.

**Filtering.** The SVC classifier assigns each unlabelled sentence a category (an intent) and a probability. We eliminate from the clusters all those sentences whose probability is below a given threshold. We then eliminate all clusters whose size is less than a set minimum.

For our subset we find the best clusterization by elbow method [**?**]. Later, having both labels for intent and cluster, we leave just that clusters that populated by items with maximum probability that allows us to cover all intent with new examples.

For each cluster we calculate number of intent representators and their overall distribution. We then keep only those clusters where some intent is represented more than in average. Elements of clusters that are left will be labelled by the majority.

**Data Expansion.** Finally, we select N labelled items for each category/intent and extend the training data with them to update sentence representation and classification 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.

# 5 Datasets

For our task we have two dataset. The first one is labelled by human and the second one was retrieved from the web to extend the previous one. Our goal was to use web data to expand the training set and improve the classifier which assigns each sentence an intent.

## 5.1 Labelled Data

The initial dataset was created manually by the Aliae startup and covers 20 main possible users intents. Each sentence represents one intent e.g.,

**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 2. 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).

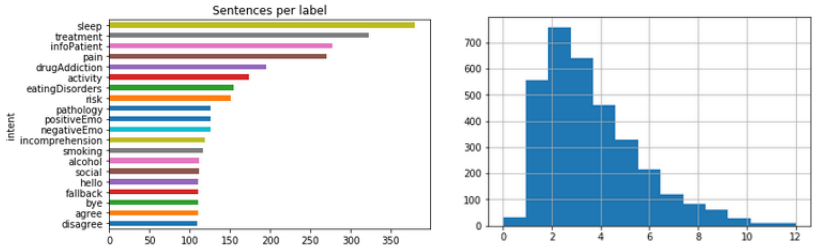


Figure 2: Left: Label distribution in dataset, Right: Distribution of sentence length

## 5.2 Forum data

To improve the initial classifier, we needed to extend the training dataset by much more data and more naturally constracted one. Looking for open medical dataset that can be used for this purpose, we found [**?**] and [**?**] works that used the HealthBoards forum data for similar tasks. HealthBoards[3] is a medical forum web portal that allows patients to discuss their ailments.

We scraped 272553 unique posts labelled with 238 forum categories. Figure 3 shows the dataset statistics. In average, sentences contain $11 \pm 7$ words after preprocessing.

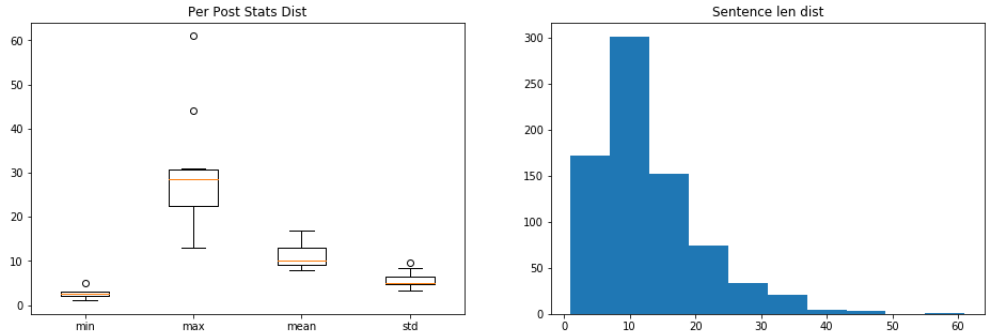


Figure 3: Text stat

This dataset has precise categorization so that some posts are more specialized and some are out of the scope of our intent labels. We should start with the data that is most similar to the initial one, so that's why we choose categories of boards that are similar to our intents. This will help us to be able gradually expand the existing context. So we will keep posts of categories provided in table 1.

[3] `healthboards.com`

| 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 1: Selected categories

Finally, the corpus consists of N sentences.

Also, for big number of iterations 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 each iteration, we only leave sentences which have mean+std words after pre-processing (cf. Section 6.1).

# 6 Experiment

Experiment consists of the main stages:

- preprocess text of labelled and unlabelled data
- use clustering to choose between different ways of representing sentences
- training a classifier on the labelled data
- iterative semi-supervised learning:
  - data expansion (cf Section 4.3) ;
  - update sentence representations (using new data and old parameters)
  - retrain on classifier on gold + new labelled data
  - evaluate new classifier on gold test data

| 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 2: Tokenization comparison

### 6.1 Preprocessing routine

The textual data was 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 2. 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 (4) looks fine.

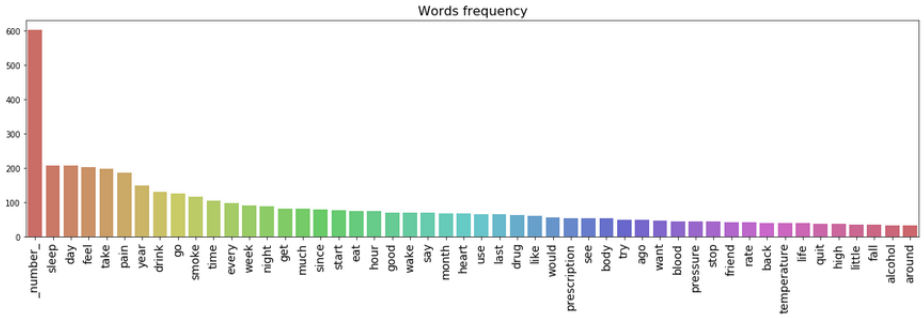


Figure 4: Words frequency

Next problem we faced is empty sentences. But there is not a lot of occurances, so their dropping don't change the dataset much.

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

These sentences are: '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"

## 6.2 Clustering for sentence representation evaluation

Thanks to clustering it is possible to see how current representation can reflect desirable labels. Also, it can provide us with information which classes will be easier to identify and which can mix with others. Using score for evalution clustering models helps us to compare different ways of representing sentences and chosing the best one.

Forum data is splitted into 20 (number of intents). 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.

### 6.2.1 Evaluation 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.

**Purity** is calculated by assigning each cluster to the class that is most often found in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N.

$$purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \tag{6}$$

where $\Omega = \{\omega_1, \omega_2, ..., \omega_K\}$ is the set of clusters and $C = \{c_1, c_2, ..., c_J\}$ is the set of classes.

**Silhouette Coefficient** shows how much the average distance to objects of its cluster differs from the average distance to objects of other clusters. This value is in the range $\lfloor 0, 1 \rfloor$. Values close to -1 correspond to poor (scattered) clustering, values close to zero indicate that the clusters intersect and overlap each other, values close to 1 correspond to "dense" clearly selected clusters. Thus, the larger the silhouette, the more clearly highlighted the clusters, and they are compact, tightly grouped clouds of points.

**Homogeneity and completeness.** Homogeneity will be maximal if the cluster consists only of objects of one class. Completeness will be maximum if all objects from the class belong to only one specific cluster. Formally, these measures are also determined using the functions of entropy and conditional entropy, considering the partitioning of the sample as discrete distributions:

$$h = 1 - \frac{H(C|K)}{H(C)}, c = 1 - \frac{H(K|C)}{H(K)} \tag{7}$$

here is $K$ the result of clustering, is $C$ the true partitioning of the sample into classes. Thus, $h$ measures how much each cluster consists of objects of one class, and $c$ measues how much objects of one class belong to one cluster.

### 6.2.2 Word2Vec model

We compare results for different embedding size ranging from 10 to 600.

| Embedding Dim. | 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 |

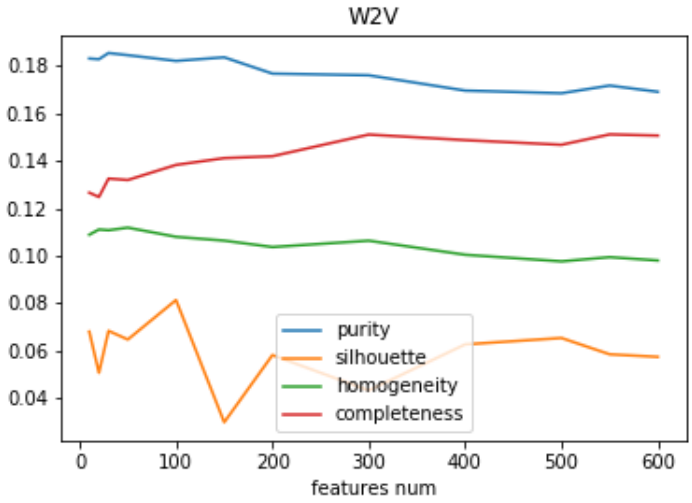


Figure 5: LDA scores

Later we will try both 10 and 100 features.

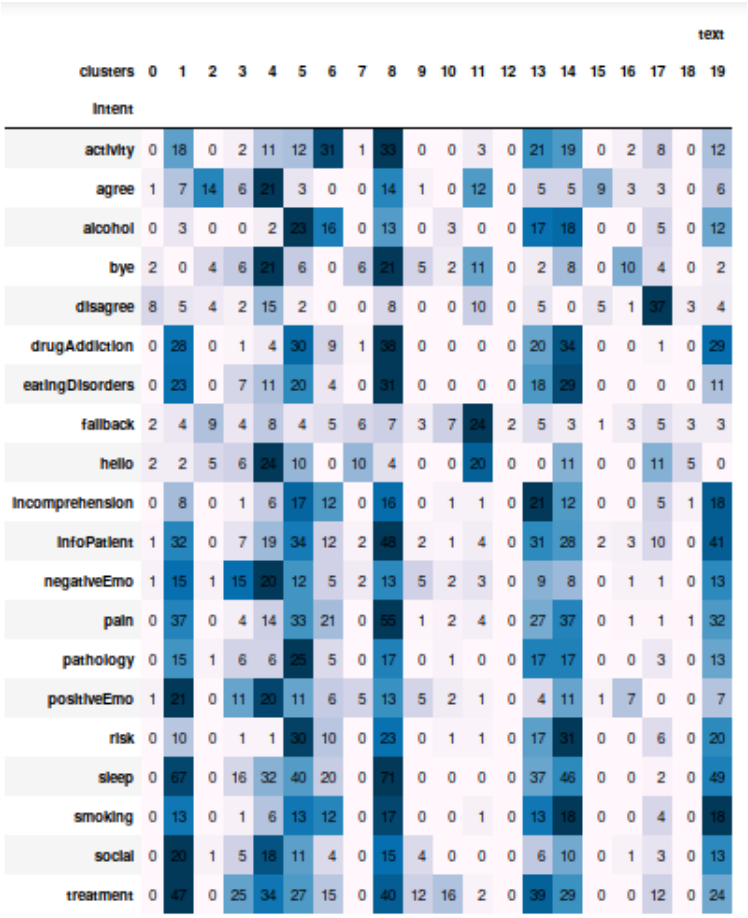


Figure 6: W2V confusion matrix

### 6.2.3 LDA model

We iterated through different numbers of topics for LDA model to see which one should be used to represent our dataset. In table 3 we can see the comparison.

| 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 | 0.166200 | 0.149270 | 0.158548 |
| 600 | 0.254261 | 0.191729 | 0.171744 | 0.186001 |

Table 3: Comparision of LDA with different number of topics

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

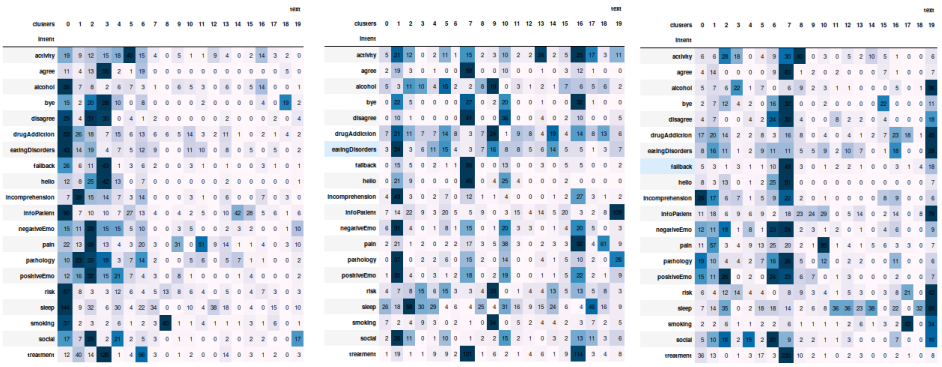

Figure 7: LDA confusion matrixes: agglomerative clustering, gaussian mixture, Kmeans

Thanks to confusion matrix that is constructed based on cluster and intent labels we can make hypotheses about possible errors and difficulties and figure out what connections our model distinguishes well. Though different clustering algorithms still capture different connections between intents, there are some common errors for all of them. In particular, the following intent are incorrectly clustered together: fallback and hello; alcohol,

drugAddiction and risk; social and emotions.

#### 6.2.4 CNN by word

From simple encoder, w2v, lda, w2v + lda

#### 6.2.5 BiLSTM by word

Following ( et al.) we trained BiLSTM encoder model verifying the best params. So we iterated through the number of layres, it's dimention and type of pooling.

| layers | purity | silhouette | homogeneity | completeness |
|---|---|---|---|---|
| 1 | **0.286369** | 0.239979 | **0.227069** | **0.232338** |
| 2 | 0.262836 | 0.181804 | 0.211068 | 0.219279 |
| 3 | 0.238284 | 0.266460 | 0.195002 | 0.206703 |
| 4 | 0.218623 | **0.439161** | 0.164478 | 0.192599 |

Table 4: Comparison of models with different number of layers

So it's better to stay with one layer. Now for one layer we test its dimension.

| layers | purity | silhouette | homogeneity | completeness |
|---|---|---|---|---|
| 32 | 0.270273 | 0.244798 | 0.214607 | 0.21474 |
| 64 | **0.281683** | 0.22728 | 0.231666 | 0.232176 |
| 128 | 0.271597 | 0.263519 | **0.234111** | **0.235486** |
| 256 | 0.270681 | 0.267634 | 0.229275 | 0.231675 |
| 512 | 0.277608 | 0.264844 | 0.231268 | 0.235075 |
| 1024 | 0.266707 | 0.260425 | 0.226291 | 0.229238 |
| 2048 | 0.268796 | **0.278391** | 0.228211 | 0.231041 |

Table 5: Dimension choosing

Finally, we chose 64 output size due to bigger difference in purity score.

Next step is chose what encoder should we use: last hiiden layer or mean-pooling or max pool on them?

| last layer | purity | silhouette | homogeneity | completeness |
|---|---|---|---|---|
| last hidden | 0.281683 | 0.22728 | 0.231666 | 0.232176 |
| mean hidden | 0.255399 | 0.170045 | 0.194848 | 0.196056 |
| max hidden | 0.272616 | 0.177766 | 0.218339 | 0.219109 |

Table 6: Choosing pooling

Finally we got BiLSTM model with one 64-dimensional layer without pooling.

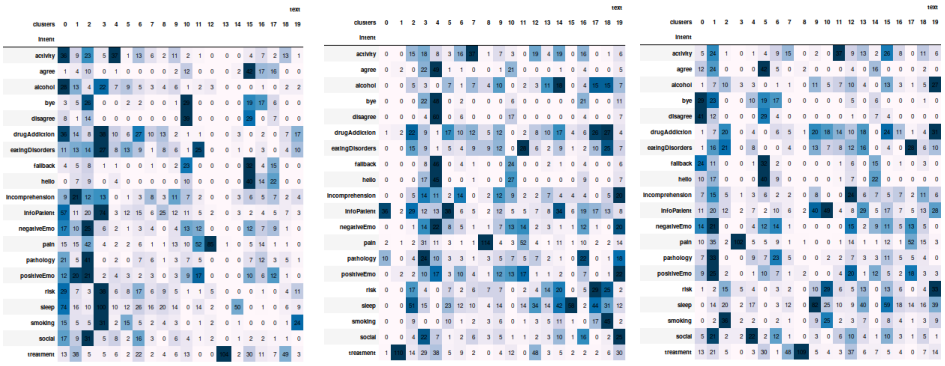

Figure 8: BiLSTM confusion matrixes: : agglomerative clustering, gaussian mixture, Kmeans

This model shows really good separatation for treatment and pain labels, tends to group together (agree, fallback and hello) and (bye and disagree).

### 6.2.6 Overall comparison

Here **??** we provide the comparision of final models taking also in consideration pretrained GloVe and Google News W2V embedding. Among word-base models LDA occured to the best one and was used for word embedding for BiLSTM as described in section 4.1. Last two models will be used for further steps.

| 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 |
| GloVe | 25 | 0.191125 | 0.003554 | 0.110058 | 0.147367 |
| GloVe | 200 | 0.186283 | -0.026413 | 0.102103 | 0.137332 |
| Google | 300 | 0.201614 | -0.016592 | 0.109450 | 0.156423 |
| LDA | 400 | 0.272718 | 0.192777 | 0.177547 | 0.192270 |
| BiLSTM | 64 | **0.281683** | 0.22728 | 0.231666 | 0.232176 |

Table 7: Overall comparison

## 6.3 Classifier

This section contains details about training classifier on the labelled data (cf. Section 5.1) and choosing the best parameters as baseline for the further semi-supervised approach.

### 6.3.1 Evaluation metrics

A classical way to evaluate a classification model performance is using accuracy score. Sometimes, especially in case of unbalanced dataset, it can be misleading because it can represent that model learns just one class. Its better to look for a precision score as a measure of exactness and a recall score as a measure of completeness.

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

where TP - True Positives, FP - False Positives, FN - False Negatives.

If we want to get an idea how our classifier deals with both classes (in simple binary case) we should use F1 score that conveys the balance between the precision and the recall.

$$F1 = \frac{2 * P * R}{P + R} \tag{10}$$

There is a more clean and unambiguous way to describe the model behaviour that is a confusion matrix. The bigger values on the main diagonal the better as well as the more balanced small values on antidiagonal.

| | Real positive | Real negative |
|---|---|---|
| Predicted positive | True Positive | False Positive |
| Predicted negative | False negative | True negative |

Table 8: Confusion matrix template

### 6.3.2 Results

For training train and test dataset were created with 65 and 43 items per category respectively that makes 60/40 ratio. Both sets are balanced, so that each category is represented by the same number of items.

SVC model was chosen with the best parameters by using Grid Search. We are comparing two kernels - rbf (C=10, gamma=1) and linear (C=10, gamma=0.001). For each model, we use 5 fold cross validation. 5-folds Cross Validation error on train set 0.401 + 0.0115 and on test set 0.371 + 0.0338. Small

| Model | Kernel | Train | Test | precision | recall | F1 |
|---|---|---|---|---|---|---|
| LDA | rbf | 0.524 + 0.00717 | 0.387 + 0.0341 | 0.308 | 0.316 | 0.287 |
| | linear | 0.371 + 0.00787 | 0.338 + 0.0431 | 0.238 | 0.241 | 0.215 |
| BiLSTM | rbf | 0.418 + 0.0193 | 0.373 + 0.0548 | 0.280 | 0.297 | 0.275 |
| | linear | 0.376 + 0.0147 | 0.343 + 0.0484 | 0.265 | 0.291 | 0.251 |

Table 9: CV score comparison

Though model with rbf model has higher scores, the linear one is less prone to overfitting as shown by the small difference between train and test errors. It was decided to try both and observe their behavior.

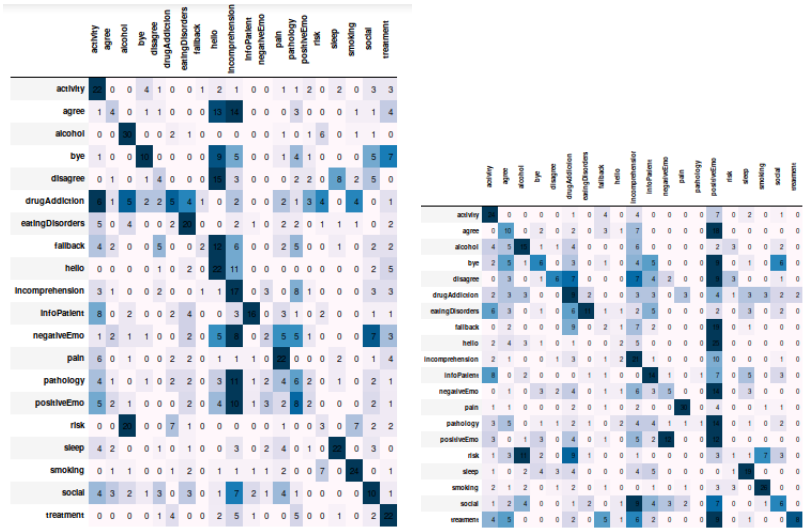


Figure 9: SVC confusion matrix

## 6.4 Semi Supervised Learning

In this section we provide classifier results obtained after each data expansion step described in 4.3.

BiLSTM model didn't succeded to provide expansion to all categories so result only for LDA model are provided.

### 6.4.1 SVC with linear kernel

On iteration 0 there is no data expansion and scores are provided for the baseline model trained on gold data.

For the first iteration we created subset of external data with $N = 500$ sentences for each predicted intent, then splitted data into 200 clusters, majority threshold=, hreshold for probability was chosen pretty low(0.12) in order to be able to extend all 20 categories. In result, we achived 15 new items.

For the second iteration we repeated the procedure choosing probability threshold as 0.1 and obtaining 13 new items.

| Iteration | Score | CV mean | CV std | precision | recall | F1 |
|---|---|---|---|---|---|---|
| 0 | 0.2651 | 0.337 | 0.0503 | 0.265 | 0.340 | 0.262 |
| 1 | 0.2895 | 0.323 | 0.0212 | 0.289 | 0.319 | 0.290 |
| 2 | 0.3174 | 0.333 | 0.0118 | 0.317 | 0.325 | 0.304 |

Table 10: Classifier scores comparison

| label | precision | | | | recall | | | | f1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| activity | 0.56 | 0.30 | 0.42 | | 0.36 | 0.27 | 0.29 | | 0.44 | 0.28 | 0.34 |
| agree | 0.23 | 0.16 | 0.05 | | 0.19 | 0.28 | 0.12 | | 0.21 | 0.21 | 0.07 |
| alcohol | 0.35 | 0.53 | 0.56 | | 0.34 | 0.44 | 0.49 | | 0.34 | 0.48 | 0.52 |
| bye | 0.14 | 0.33 | 0.53 | | 0.24 | 0.41 | 0.38 | | 0.18 | 0.36 | 0.44 |
| disagree | 0.14 | 0.14 | 0.37 | | 0.40 | 0.08 | 0.38 | | 0.21 | 0.10 | 0.38 |
| drugAddiction | 0.21 | 0.21 | 0.23 | | 0.12 | 0.29 | 0.23 | | 0.15 | 0.24 | 0.23 |
| eatingDisorders | 0.26 | 0.37 | 0.28 | | 0.61 | 0.73 | 0.34 | | 0.36 | 0.49 | 0.31 |
| fallback | 0.05 | 0.12 | 0.00 | | 0.09 | 0.31 | 0.00 | | 0.06 | 0.17 | 0.00 |
| hello | 0.05 | 0.00 | 0.21 | | 0.17 | 0.00 | 0.43 | | 0.07 | 0.00 | 0.28 |
| incomprehension | 0.49 | 0.37 | 0.35 | | 0.20 | 0.26 | 0.20 | | 0.29 | 0.31 | 0.25 |
| infoPatient | 0.33 | 0.49 | 0.49 | | 0.25 | 0.34 | 0.44 | | 0.29 | 0.40 | 0.46 |
| negativeEmo | 0.12 | 0.05 | 0.16 | | 0.21 | 0.06 | 0.21 | | 0.15 | 0.05 | 0.18 |
| pain | 0.70 | 0.60 | 0.70 | | 0.81 | 0.46 | 0.81 | | 0.75 | 0.53 | 0.75 |
| pathology | 0.02 | 0.05 | 0.05 | | 0.50 | 0.09 | 0.29 | | 0.04 | 0.06 | 0.08 |
| positiveEmo | 0.28 | 0.37 | 0.09 | | 0.07 | 0.11 | 0.14 | | 0.11 | 0.17 | 0.11 |
| risk | 0.02 | 0.12 | 0.07 | | 0.08 | 0.16 | 0.11 | | 0.04 | 0.13 | 0.08 |
| sleep | 0.44 | 0.42 | 0.49 | | 0.47 | 0.50 | 0.54 | | 0.46 | 0.46 | 0.51 |
| smoking | 0.60 | 0.44 | 0.49 | | 0.68 | 0.59 | 0.66 | | 0.64 | 0.51 | 0.56 |
| social | 0.14 | 0.14 | 0.26 | | 0.20 | 0.21 | 0.31 | | 0.16 | 0.17 | 0.28 |
| treatment | 0.19 | 0.58 | 0.56 | | 0.80 | 0.75 | 0.15 | | 0.30 | 0.67 | 0.24 |

Table 11: Comparision of classification scores for SVC with linear kernel

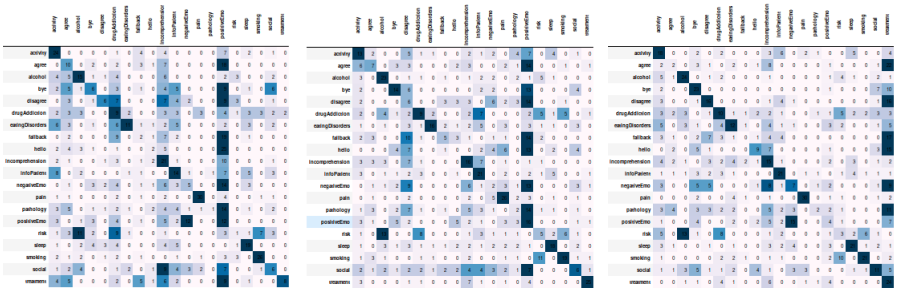

Figure 10: SVC confusion matrix for iteration 0, 1, 2.

### 6.4.2 SVC with rbf kernel

Zero iteration stands for the baseline model performance.

For the first iteration we created subset of external data with $N = 500$ sentences for each predicted intent, then splitted data into 1000 clusters, hreshold for probability was chosen pretty low (0.15) in order to be able to extend all 20 categories. In result, we achived 6 new items.

For the second iteration we used the same parameters, threshold for probability became lower(0.11) and in result 5 new items were obtained.

Here we provide grouped results about classifier scores chaging by iteration.

| Iteration | Score | CV mean | CV std | precision | recall | F1 |
|---|---|---|---|---|---|---|
| 0 | 0.3186 | 0.371 | 0.0338 | 0.308 | 0.316 | 0.287 |
| 1 | 0.3523 | 0.350 | 0.0153 | 0.352 | 0.371 | 0.346 |
| 2 | 0.3167 | 0.332 | 0.0150 | 0.316 | 0.331 | 0.313 |

Table 12: Classifier scores comparison

| label | precision | | | | recall | | | | f1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | it 0 | it 1 | it 2 | | it 0 | it 1 | it 2 | | it 0 | it 1 | it 2 |
| activity | 0.44 | 0.49 | 0.37 | | 0.40 | 0.54 | 0.34 | | 0.42 | 0.51 | 0.36 |
| agree | 0.72 | 0.23 | 0.14 | | 0.20 | 0.31 | 0.21 | | 0.32 | 0.27 | 0.17 |
| alcohol | 0.51 | 0.56 | 0.47 | | 0.44 | 0.41 | 0.39 | | 0.47 | 0.47 | 0.43 |
| bye | 0.35 | 0.56 | 0.33 | | 0.29 | 0.67 | 0.30 | | 0.32 | 0.39 | 0.31 |
| disagree | 0.33 | 0.33 | 0.35 | | 0.38 | 0.31 | 0.43 | | 0.35 | 0.32 | 0.38 |
| drugAddiction | 0.09 | 0.26 | 0.28 | | 0.13 | 0.39 | 0.29 | | 0.11 | 0.31 | 0.28 |
| eatingDisorders | 0.33 | 0.47 | 0.33 | | 0.45 | 0.67 | 0.25 | | 0.38 | 0.52 | 0.28 |
| fallback | 0.00 | 0.26 | 0.14 | | 0.00 | 0.12 | 0.12 | | 0.00 | 0.17 | 0.13 |
| hello | 0.23 | 0.51 | 0.40 | | 0.25 | 0.25 | 0.19 | | 0.24 | 0.34 | 0.26 |
| incomprehension | 0.51 | 0.30 | 0.42 | | 0.29 | 0.29 | 0.43 | | 0.37 | 0.30 | 0.42 |
| infoPatient | 0.30 | 0.42 | 0.37 | | 0.54 | 0.35 | 0.52 | | 0.39 | 0.38 | 0.43 |
| negativeEmo | 0.12 | 0.05 | 0.14 | | 0.19 | 0.06 | 0.18 | | 0.14 | 0.05 | 0.16 |
| pain | 0.30 | 0.65 | 0.49 | | 0.22 | 0.49 | 0.44 | | 0.25 | 0.56 | 0.46 |
| pathology | 0.09 | 0.14 | 0.16 | | 0.16 | 0.35 | 0.29 | | 0.12 | 0.20 | 0.21 |
| positiveEmo | 0.09 | 0.14 | 0.16 | | 0.17 | 0.20 | 0.20 | | 0.12 | 0.16 | 0.18 |
| risk | 0.05 | 0.07 | 0.05 | | 0.17 | 0.14 | 0.11 | | 0.07 | 0.09 | 0.06 |
| sleep | 0.42 | 0.58 | 0.56 | | 0.58 | 0.66 | 0.63 | | 0.49 | 0.62 | 0.59 |
| smoking | 0.65 | 0.51 | 0.58 | | 0.61 | 0.54 | 0.54 | | 0.63 | 0.52 | 0.56 |
| social | 0.33 | 0.19 | 0.21 | | 0.50 | 0.29 | 0.56 | | 0.49 | 0.23 | 0.31 |
| treatment | 0.51 | 0.63 | 0.40 | | 0.35 | 0.40 | 0.22 | | 0.42 | 0.49 | 0.28 |

Table 13: Comparision of classification scores for SVC with rbf kernel

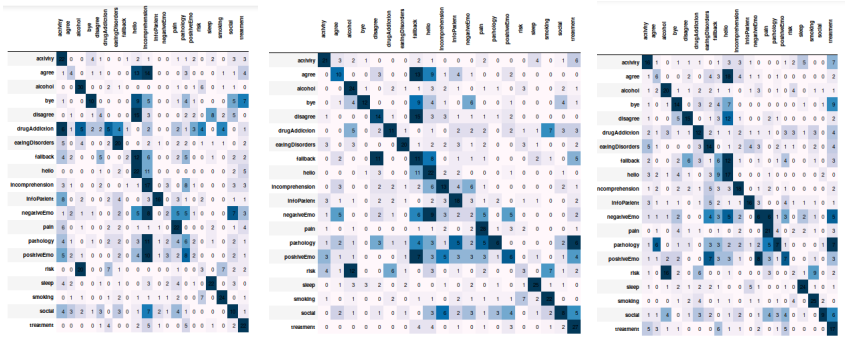

Figure 11: SVC confusion matrix for iteration 0, 1, 2.

# 7 Conclusion and future work

In this project we were dealing with task of paraphrase detection for expanding initial dataset. Our approch is to train the word embedding model that will be able to provide clusters similar to initial classes. We used LDA and W2V models for that and later trained SVC and Kmeans models to label external data and chose the representative example to extend existing dataset and retrain models on it.

Our results shows how high rate of confusion between labels in intial classifier increases this confusion in futher data expansion though in general algorithms tends to converge to be more precise. Also, the more data you have the higher chances to find more similar sentences to initial one so that possible context for each intent will grow gradually.

Future work will be devoted to improving word embedding model, trying one-shot learning model in order to improve classification score. Transfering from one-intent to multi-intent is preferable due to the fact that sentences are usually ambiguous than not. Another possible direction for development can be creating framework for adding new labels to the existing classifier.

# 8 References

$WardJ.H.Hierarchicalgroupingtooptimizeanobjectivefunction//J.oftheAmericanStatisticalAssoci$

# 9 Annex

## 9.1 Semi linear

| label | precision | recall | f1-score | support |
|---|---|---|---|---|
| | | | | |
| activity | 0.56 | 0.36 | 0.44 | 66 |
| agree | 0.23 | 0.19 | 0.21 | 54 |
| alcohol | 0.35 | 0.34 | 0.34 | 44 |
| bye | 0.14 | 0.24 | 0.18 | 25 |
| disagree | 0.14 | 0.40 | 0.21 | 15 |
| drugAddiction | 0.21 | 0.12 | 0.15 | 75 |
| eatingDisorders | 0.26 | 0.61 | 0.36 | 18 |
| fallback | 0.05 | 0.09 | 0.06 | 22 |
| hello | 0.05 | 0.17 | 0.07 | 12 |
| incomprehension | 0.49 | 0.20 | 0.29 | 103 |
| infoPatient | 0.33 | 0.25 | 0.29 | 55 |
| negativeEmo | 0.12 | 0.21 | 0.15 | 24 |
| pain | 0.70 | 0.81 | 0.75 | 37 |
| pathology | 0.02 | 0.50 | 0.04 | 2 |
| positiveEmo | 0.28 | 0.07 | 0.11 | 178 |
| risk | 0.02 | 0.08 | 0.04 | 12 |
| sleep | 0.44 | 0.47 | 0.46 | 40 |
| smoking | 0.60 | 0.68 | 0.64 | 38 |
| social | 0.14 | 0.20 | 0.16 | 30 |
| treatment | 0.19 | 0.80 | 0.30 | 10 |
| | | | | |
| micro avg | 0.27 | 0.27 | 0.27 | 860 |
| macro avg | 0.27 | 0.34 | 0.26 | 860 |
| weighted avg | 0.33 | 0.27 | 0.27 | 860 |

Table 14: Classifier 0

| label | precision | recall | f1-score | support |
|---|---|---|---|---|
| | | | | |
| activity | 0.30 | 0.27 | 0.28 | 49 |
| agree | 0.16 | 0.28 | 0.21 | 25 |
| alcohol | 0.53 | 0.44 | 0.48 | 52 |
| bye | 0.33 | 0.41 | 0.36 | 34 |
| disagree | 0.14 | 0.08 | 0.10 | 78 |
| drugAddiction | 0.21 | 0.29 | 0.24 | 31 |
| eatingDisorders | 0.37 | 0.73 | 0.49 | 22 |
| fallback | 0.12 | 0.31 | 0.17 | 16 |
| hello | 0.00 | 0.00 | 0.00 | 19 |
| incomprehension | 0.37 | 0.26 | 0.31 | 61 |
| infoPatient | 0.49 | 0.34 | 0.40 | 61 |
| negativeEmo | 0.05 | 0.06 | 0.05 | 31 |
| pain | 0.60 | 0.46 | 0.53 | 56 |
| pathology | 0.05 | 0.09 | 0.06 | 23 |
| positiveEmo | 0.37 | 0.11 | 0.17 | 142 |
| risk | 0.12 | 0.16 | 0.13 | 32 |
| sleep | 0.42 | 0.50 | 0.46 | 36 |
| smoking | 0.44 | 0.59 | 0.51 | 32 |
| social | 0.14 | 0.21 | 0.17 | 28 |
| treatment | 0.58 | 0.78 | 0.67 | 32 |
| | | | | |
| micro avg | 0.29 | 0.29 | 0.29 | 860 |
| macro avg | 0.29 | 0.32 | 0.29 | 860 |
| weighted avg | 0.33 | 0.29 | 0.29 | 860 |

Table 15: Classifier 1

| label | precision | recall | f1-score | support |
|---|---|---|---|---|
| | | | | |
| activity | 0.42 | 0.29 | 0.34 | 63 |
| agree | 0.05 | 0.12 | 0.07 | 17 |
| alcohol | 0.56 | 0.49 | 0.52 | 49 |
| bye | 0.53 | 0.38 | 0.44 | 61 |
| disagree | 0.37 | 0.38 | 0.38 | 42 |
| drugAddiction | 0.23 | 0.23 | 0.23 | 43 |
| eatingDisorders | 0.28 | 0.34 | 0.31 | 35 |
| fallback | 0.00 | 0.00 | 0.00 | 6 |
| hello | 0.21 | 0.43 | 0.28 | 21 |
| incomprehension | 0.35 | 0.20 | 0.25 | 76 |
| infoPatient | 0.49 | 0.44 | 0.46 | 48 |
| negativeEmo | 0.16 | 0.21 | 0.18 | 34 |
| pain | 0.70 | 0.81 | 0.75 | 37 |
| pathology | 0.05 | 0.29 | 0.08 | 7 |
| positiveEmo | 0.09 | 0.14 | 0.11 | 29 |
| risk | 0.07 | 0.11 | 0.08 | 28 |
| sleep | 0.49 | 0.54 | 0.51 | 39 |
| smoking | 0.49 | 0.66 | 0.56 | 32 |
| social | 0.26 | 0.31 | 0.28 | 36 |
| treatment | 0.56 | 0.15 | 0.24 | 157 |
| | | | | |
| micro avg | 0.32 | 0.32 | 0.32 | 860 |
| macro avg | 0.32 | 0.32 | 0.30 | 860 |
| weighted avg | 0.40 | 0.32 | 0.33 | 860 |

Table 16: Classifier 2

### 9.2 Semi rbf

| label | precision | recall | f1-score | support |
|---|---|---|---|---|
| activity | 0.44 | 0.40 | 0.42 | 47 |
| agree | 0.72 | 0.20 | 0.32 | 152 |
| alcohol | 0.51 | 0.44 | 0.47 | 50 |
| bye | 0.35 | 0.29 | 0.32 | 51 |
| disagree | 0.33 | 0.38 | 0.35 | 37 |
| drugAddiction | 0.09 | 0.13 | 0.11 | 30 |
| eatingDisorders | 0.33 | 0.45 | 0.38 | 31 |
| fallback | 0.00 | 0.00 | 0.00 | 8 |
| hello | 0.23 | 0.25 | 0.24 | 40 |
| incomprehension | 0.51 | 0.29 | 0.37 | 76 |
| infoPatient | 0.30 | 0.54 | 0.39 | 24 |
| negativeEmo | 0.12 | 0.19 | 0.14 | 26 |
| pain | 0.30 | 0.22 | 0.25 | 59 |
| pathology | 0.09 | 0.16 | 0.12 | 25 |
| positiveEmo | 0.09 | 0.17 | 0.12 | 24 |
| risk | 0.05 | 0.17 | 0.07 | 12 |
| sleep | 0.42 | 0.58 | 0.49 | 31 |
| smoking | 0.65 | 0.61 | 0.63 | 46 |
| social | 0.33 | 0.50 | 0.39 | 28 |
| treatment | 0.51 | 0.35 | 0.42 | 63 |
| | | | | |
| micro avg | 0.32 | 0.32 | 0.32 | 860 |
| macro avg | 0.32 | 0.32 | 0.30 | 860 |
| weighted avg | 0.42 | 0.32 | 0.34 | 860 |

Table 17: Classification report for 0 iteration

| label | precision | recall | f1-score | support |
|---|---|---|---|---|
| | | | | |
| activity | 0.49 | 0.54 | 0.51 | 39 |
| agree | 0.23 | 0.31 | 0.27 | 32 |
| alcohol | 0.56 | 0.41 | 0.47 | 59 |
| bye | 0.28 | 0.67 | 0.39 | 18 |
| disagree | 0.33 | 0.31 | 0.32 | 45 |
| drugAddiction | 0.26 | 0.39 | 0.31 | 28 |
| eatingDisorders | 0.47 | 0.67 | 0.55 | 30 |
| fallback | 0.26 | 0.12 | 0.17 | 88 |
| hello | 0.51 | 0.25 | 0.34 | 88 |
| incomprehension | 0.30 | 0.29 | 0.30 | 45 |
| infoPatient | 0.42 | 0.35 | 0.38 | 52 |
| negativeEmo | 0.05 | 0.06 | 0.05 | 36 |
| pain | 0.65 | 0.49 | 0.56 | 57 |
| pathology | 0.14 | 0.35 | 0.20 | 17 |
| positiveEmo | 0.14 | 0.20 | 0.16 | 30 |
| risk | 0.07 | 0.14 | 0.09 | 21 |
| sleep | 0.58 | 0.66 | 0.62 | 38 |
| smoking | 0.51 | 0.54 | 0.52 | 41 |
| social | 0.19 | 0.29 | 0.23 | 28 |
| treatment | 0.63 | 0.40 | 0.49 | 68 |
| | | | | |
| micro avg | 0.35 | 0.35 | 0.35 | 860 |
| macro avg | 0.35 | 0.37 | 0.35 | 860 |
| weighted avg | 0.40 | 0.35 | 0.36 | 860 |

Table 18: Classification report for the first iteration

| label | precision | recall | f1-score | support |
|---|---|---|---|---|
| | | | | |
| activity | 0.37 | 0.34 | 0.36 | 47 |
| agree | 0.14 | 0.21 | 0.17 | 29 |
| alcohol | 0.47 | 0.39 | 0.43 | 51 |
| bye | 0.33 | 0.30 | 0.31 | 46 |
| disagree | 0.35 | 0.43 | 0.38 | 35 |
| drugAddiction | 0.28 | 0.29 | 0.28 | 42 |
| eatingDisorders | 0.33 | 0.25 | 0.28 | 57 |
| fallback | 0.14 | 0.12 | 0.13 | 51 |
| hello | 0.40 | 0.19 | 0.26 | 90 |
| incomprehension | 0.42 | 0.43 | 0.42 | 42 |
| infoPatient | 0.37 | 0.52 | 0.43 | 31 |
| negativeEmo | 0.14 | 0.18 | 0.16 | 34 |
| pain | 0.49 | 0.44 | 0.46 | 48 |
| pathology | 0.16 | 0.29 | 0.21 | 24 |
| positiveEmo | 0.16 | 0.20 | 0.18 | 35 |
| risk | 0.05 | 0.11 | 0.06 | 19 |
| sleep | 0.56 | 0.63 | 0.59 | 38 |
| smoking | 0.58 | 0.54 | 0.56 | 46 |
| social | 0.21 | 0.56 | 0.31 | 16 |
| treatment | 0.40 | 0.22 | 0.28 | 79 |
| | | | | |
| micro avg | 0.32 | 0.32 | 0.32 | 860 |
| macro avg | 0.32 | 0.33 | 0.31 | 860 |
| weighted avg | 0.34 | 0.32 | 0.32 | 860 |
| | | | | |

Table 19: Classification report for the second iteration

| Intent | It 0 | It 1 | It 2 |
|---|---|---|---|
| activity | 0.31+0.10 | 0.51+0.02 | |
| agree | 0.17+0.06 | 0.32+0.01 | |
| alcohol | 0.37+0.13 | 0.53+0.03 | |
| bye | 0.17+0.03 | 0.40+0.03 | |
| disagree | 0.19+0.03 | 0.27+0.03 | |
| drugAddiction | 0.22+0.06 | 0.32+0.03 | |
| eatingDisorders | 0.33+0.13 | 0.66+0.07 | |
| fallback | 0.16+0.03 | 0.19+0.05 | |
| hello | 0.17+0.02 | 0.31+0.14 | |
| incomprehension | 0.20+0.04 | 0.46+0.02 | |
| infoPatient | 0.39+0.18 | 0.71+0.05 | |
| negativeEmo | 0.17+0.03 | 0.34+0.06 | |
| pain | 0.36+0.14 | 0.67+0.05 | |
| pathology | 0.16+0.04 | 0.39+0.06 | |
| positiveEmo | 0.19+0.04 | 0.27+0.03 | |
| risk | 0.28+0.07 | 0.31+0.03 | |
| sleep | 0.42+0.18 | 0.96+0.01 | |
| smoking | 0.49+0.17 | 0.79+0.02 | |
| social | 0.23+0.07 | 0.38+0.04 | |
| treatment | 0.25+0.04 | 0.43+0.04 | |

Table 20: Intention mean probability

| Intent | It 0 | It 1 | It 2 |
|---|---|---|---|
| activity | 0.31+0.10 | 0.51+0.02 | |
| agree | 0.17+0.06 | 0.32+0.01 | |
| alcohol | 0.37+0.13 | 0.53+0.03 | |
| bye | 0.17+0.03 | 0.40+0.03 | |
| disagree | 0.19+0.03 | 0.27+0.03 | |
| drugAddiction | 0.22+0.06 | 0.32+0.03 | |
| eatingDisorders | 0.33+0.13 | 0.66+0.07 | |
| fallback | 0.16+0.03 | 0.19+0.05 | |
| hello | 0.17+0.02 | 0.31+0.14 | |
| incomprehension | 0.20+0.04 | 0.46+0.02 | |
| infoPatient | 0.39+0.18 | 0.71+0.05 | |
| negativeEmo | 0.17+0.03 | 0.34+0.06 | |
| pain | 0.36+0.14 | 0.67+0.05 | |
| pathology | 0.16+0.04 | 0.39+0.06 | |
| positiveEmo | 0.19+0.04 | 0.27+0.03 | |
| risk | 0.28+0.07 | 0.31+0.03 | |
| sleep | 0.42+0.18 | 0.96+0.01 | |
| smoking | 0.49+0.17 | 0.79+0.02 | |
| social | 0.23+0.07 | 0.38+0.04 | |
| treatment | 0.25+0.04 | 0.43+0.04 | |

Table 21: Intention mean probability