UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Detection of Difficult for Understanding Medical Words using Deep Learning

Author:
Hanna PYLIEVA

Supervisor:
PhD. Artem CHERNODUB

Co-supervisors:
PhD. Natalia GRABAR
PhD. Thierry HAMON

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2019

Declaration of Authorship

I, Hanna PYLIEVA, declare that this thesis titled, "Detection of Difficult for Understanding Medical Words using Deep Learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UKRAINIAN CATHOLIC UNIVERSITY

Abstract

Faculty of Applied Sciences

Master of Science

Detection of Difficult for Understanding Medical Words using Deep Learning

by Hanna PYLIEVA

In the medical domain, non-specialized users often require a better understanding of medical information provided by doctors. In this work, we address this need. We introduce novel embeddings received from RNN - FrnnMUTE (French RNN Medical Understandability Text Embeddings) - and show how they help to improve identification of readability and understandability of medical words when applied as features in the classification task, reaching at maximum 87.0 F1 score. We also found out that adding pre-trained FastText word embeddings to the feature set substantially improves the performance of the classification model. For generalizability study of different models, we introduce a methodology comprising three cross-validation scenarios which allow testing classifiers in real-world conditions: when understanding of medical words by new users is unknown or when no information about understandability of new words is provided for the model.

Acknowledgements

First of all, I would like to thank my supervisor Artem Chernodub (Ukrainian Catholic University, Grammarly) who directed me throughout research for this thesis and provided a lot of useful pieces of advice regarding contents, structure and possible future development of the project.

Also, I would like to thank Natalia Grabar (CNRS at Université de Lille, France), and Thierry Hamon (LIMSI, CNRS at Université Paris-Saclay, France and Université Paris 13, France) who agreed to proceed the work on detection of understandability of medical words, provided data for the research and actively participated in discussions on the project's progress and preparation of publications.

Finally, I am grateful to Ukrainian Catholic University and Oleksii Molchanovskyi personally for the Master Program which was an important step in my career development as a data scientist.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	**A**rtificial **N**eural **N**etwork
CV	**C**ross-**v**alidation
CWI	**C**omplex **W**ord **I**dentification
DNN	**D**eep **N**eural **N**etwork
DT	**D**ecision **T**ree
FrnnMute	**F**rench **r**ecurrent **n**eural **n**etwork **M**edical **U**nderstability **T**ext **E**mbeddings
GPU	**G**raphics **P**rocessing **U**nit
NLP	**N**atural **L**anguage **P**rocessing
RNN	**R**ecurrent **N**eural **N**etwork
LSTM	**L**ong **S**hort-**T**erm **M**emory

List of Symbols

A accuracy
P precision
R recall
F F_1 score
μ mean of a sample
σ standard deviation of a sample

Chapter 1

Introduction

1.1 Motivation

Specialized areas, such as the medical area, convey and use technical words or terms which are typically related to knowledge developed within these areas. In the medical area, this specific knowledge often corresponds to fundamental medical notions related to disorders, procedures, treatments and human anatomy. For instance, technical terms like *blepharospasm* (abnormal contraction or twitch of the eyelid), *alexithymia* (inability to identify and describe emotions in the self), *appendicectomy* (surgical removal of the vermiform appendix from intestine), or *lombalgia* (low back pain) are frequently used by experts in medical texts.

As in any specialized area, two main kinds of users exist in the medical area:

- experts of the domain: medical doctors, both researchers and practitioners. They contribute to the creation and development of biomedical knowledge and its presentation for the healthcare process of patients;
- consumers of the healthcare process: patients and their relatives. Usually, they do not have expert knowledge, while it is important that they understand the purpose and issues of their healthcare process.

One more intermediate group of medical area users can be specified: users who are not experts in the area but have some knowledge of the medical domain. In (Pearson, 1998) this group of users is named "initiates". Users of this group are either in the learning process (students) or do not need more detailed knowledge in the medical domain (technicians). Initiates and medical doctors form a group of *medical stuff* - users who do not have difficulties in understanding technical medical terms. On the contrary, patients and their relatives may find it difficult to understand and use such terms. This group of users shows poor *health literacy*.

The existing literature provides several studies dedicated to the understanding of medical notions and terms by non-expert users, and how the level of health literacy of patients impacts on a successful healthcare process (McCray, 2005; Eysenbach, 2007). It is not uncommon that patients and their relatives must face very technical health documents and information. Examples of this kind are frequent, and usually, the non-expert users are at a loss in such situations:

- understanding information on drug intake (Vander Stichele, 1999; Patel, Branch, and Arocha, 2002), such as instructions related to the description and specification of steps necessary for the preparation and intake of drugs,
- understanding clinical documents (Zeng-Treiler et al., 2007) which contain important information on the healthcare process of patients,

- understanding clinical brochures or informed consents (Williams et al., 1995) which are specifically created for patients and which are typically read by patients during their clinical pathway,
- more generally, understanding the information provided for patients by different websites (Oregon Practice Center, 2008; Brigo et al., 2015) in different languages (English, Spanish, French) and different medical specialties,
- for the same reasons, communication between patients and medical staff (Jucks and Bromme, 2007; Tran et al., 2009) remains complicated.

These various observations provide the main motivation for our work. In this work, we address the needs of non-specialized users in the medical domain. As we noticed, the main need is related to the understanding of medical and health information.

The recent increase of availability of medical data and the rapid spread of big data analytics tools have facilitated the broad application of deep learning techniques in the healthcare domain (Jiang et al., 2017). The popularity of such methods is due to their ability to mine features 'on the go' from massive datasets of any type (either table, text, image or audio) and produce valuable insights. Although classical analytics and machine learning approaches require less data for learning patterns, they need a set of features representing the dataset which need to be engineered before the learning process. Feature engineering in its turn often involves deep domain understanding and moreover becomes a time-consuming process, whereas the results of learning on such features nowadays are mostly weaker in tasks of computer vision and natural language processing (Krizhevsky, Sutskever, and Hinton, 2012; Zhang et al., 2015).

1.2 The proposed method

Taking into consideration all of the above, we propose the following:

- applying deep learning techniques for better identification of readability and understandability of medical words by non-expert users. In particular, we will solve a words' categorization task and compare the performance of a classification model on different feature sets: standard linguistic and non-linguistic features described in chapter 5, ones obtained using different deep learning approaches and combinations of the previous two.
- investigating how different feature sets perform with three different cross-validation settings, described in chapter 6.

The medical data used in this work are in French. Seven human annotators participated in the creation of the reference data (labels specifying understandability of words).

1.3 Goals of the master thesis

1. To provide an overview of previous works on word understandability detection.

2. To apply deep learning techniques for a generation of word features used then by a word categorization[1] on understandable and not by non-specialists.

3. To compare the quality of word categorization from the perspective of understandability on different sets of features and explain the causes of differences in performance.

1.4 Thesis structure

We first present some related work to our task in chapter 2. In chapter 3 we provide background information which forms the basis of methods we propose and describe later in chapter 5. In chapter 4 we introduce the data used throughout this work. Our results of applying the proposed methods are presented and discussed in chapter 6. Finally, we summarize our contributions and list the directions for future work in chapter 7.

[1] We will use the words '*categorization*' and '*classification*' interchangeably in this work, implying that in the scope of our task these words are synonyms, whereas the first one is common in the medical domain, and the second one - in machine learning.

Chapter 2

Related Work

Related work is globally related to the text simplification task which involves the detection of complex contents in documents and their adaptation for the target population. In this work, we are interested in the first aspect with additional constraints: detection and diagnosis of technical contents in texts of medical domain. In general non-domain specific context, this task is also known in the literature as complex words identification (CWI). From the overview of related works, it will be clear that in the NLP (Natural Language Processing) area, work related to the diagnosis of technical content in general and in the medical domain, in particular, is quite frequent and topical.

2.1 Early research in readability measurement

Readability is the ease of understanding written text. The study of readability and how it can be measured takes origin from 1880th with analytics of literature and poetry (Sherman, 1893). Then, *traditional readability measures* were invented. They rely on two main factors: the familiarity of semantic units such as words or phrases, and the complexity of syntax. Due to the intention of making these measures straightforward for applications, some simplifying assumptions were used. As a result, final formulas mostly rely on the number of letters and/or of syllables a word contains and on linear regression models (Flesch, 1948; Gunning, 1973). While such readability measures are easy to compute, they are based on shallow characteristics of text, ignoring deeper levels of text processing which are important factors in readability, such as cohesion, syntactic ambiguity, rhetorical organization, and propositional density (Collins-Thompson, 2014). Moreover, traditional measures of readability were demonstrated to be unreliable for Web pages and other types of non-traditional documents during the recent studies (Si and P. Callan, 2001). As a result of such limitations and due to the recent growth of computational and data resources, researchers in Natural Language Processing (NLP) area started to work on *computational* readability measurements, which relies on the use of machine learning algorithms on richer linguistic features.

2.2 Data sources

Machine learning-based approaches require suitable data to produce accurate and usable models. Creation of data sources for CWI is a special and separate field of study. In recent years, several approaches have been proposed:

- use of expert judgment, who have an idea on needs of population aimed in the study (Clercq et al., 2014). The main limitation is that experts may have difficulties in figuring out what are the real needs of the population;

- use of textbooks created for the population according to their readability levels, such as school books (Gala, François, and Fairon, 2013). The main limitation is that such books are usually created by experts using a theoretical basis and observations;
- use of crowdsourcing involving a large group (Clercq et al., 2014). The main limitation is that the group involved is uncontrolled and unknown;
- use of eye-tracking methods for a more fine-grained analysis of reading difficulties (Yaneva, Temnikova, and Mitkov, 2015; Grabar, Farce, and Sparrow, 2018). The main limitation is that only short text spans can be used;
- manual annotation by human annotators (Grabar and Hamon, 2016). In this case, the annotators represent the population; they are part of the controlled population, they can perform more complicated tasks than in case of crowdsourcing, although they are usually less than in crowdsourcing experiments. In this work, the data source was constructed using this method. It also was exploited in CWI challenges mentioned in the next section (SemEval-2016 and CWI 2018 Shared Task).

Related to this issue is the question on the generalizability of data and models generated from these data. For instance, it has been observed that data from experts are difficult to generalize over the population (Clercq et al., 2014).

2.3 Automated readability assessment

2.3.1 General language

For general language, research actions are often performed as a part of NLP challenges. For the case of CWI for example, there was a shared task on CWI on SemEval-2016 NLP challenge[1]. The goal was to provide a framework for the evaluation of CWI methods, which involved:

1. understanding the distinctive characteristics of words which are difficult for non-native speakers;
2. finding out how well the vocabulary limitations of an individual can be predicted from the knowledge of vocabulary limitations of the group they are part of;
3. introducing a gold-standard dataset for text simplification and tasks related to topic modeling and semantics.

The participants applied rule-based and/or machine learning systems, including neural networks for building solutions. Combinations of various features, designed to detect the complexity of words, have been used. The most popular among them were:

- simple features: word length, number of syllables, named-entity type, part-of-speech, the position of a word in sentence (Bingel, Schluter, and Martínez Alonso, 2016);

[1] http://alt.qcri.org/semeval2016/task11/

- number of synsets, senses, synonyms, hyponyms, relations, distinct POSs in WordNet (Ronzano et al., 2016);
- corpus-based frequency in large corpora: Wikipedia, Simple Wikipedia (Kauchak, 2013), SubIMDB (Paetzold and Specia, 2016b), British National Corpus (Ronzano et al., 2016), Gigaword corpus and the International Conference on Web and Social Media (ICWSM) blog corpus (Brooke, Uitdenbogerd, and Baldwin, 2016). Mostly the frequency was calculated for word-level, but some participants utilized the frequency of char-level n-grams as well (Bingel, Schluter, and Martínez Alonso, 2016).

The results of this shared task are described in detail in Paetzold and Specia, 2016a. The analysis of 42 submitted systems by 21 teams highlighted that the most effectively CWI task is solved using decision trees (Malmasi, Dras, and Zampieri, 2016) and ensemble methods (Paetzold and Specia, 2016b; Ronzano et al., 2016). Moreover, according to the results, word frequencies remained the most reliable predicting feature of word complexity. The best systems reached up to 77.4 G-score, which measures the harmonic mean between Accuracy and Recall, and 35.3 F-score.

In this challenge, attempts to apply neural networks showed poor results. Whereas after post-task experiments authors gained competitive results changing the framework of NN implementation, revising architecture and the feature set (Bingel, Schluter, and Martínez Alonso, 2016). Among features, 300-dimensional GloVe[2] word embeddings were found to be the main contributor to NN's performance improvement (from 50.6 to 75.6 G-score).

Our task is slightly different from the one described in SemEval-2016 Shared task (Paetzold and Specia, 2016a) where given a sentence and a target word within it, the goal is to predict whether or not a non-native English speaker would be able to understand the meaning of the target word. In our formulation we do not have the context near target medical words, so we cannot use it during the training. In other words, the task in SemEval-2016 is CWI in its ordinary meaning, whereas in our case the task comes down to words' classification. The usefulness of standard word embeddings for our task is also not clear, therefore. Moreover, in SemEval-2016 and our task user annotations are made in different languages: English and French correspondingly, - and have different goals.

After the success of SemEval-2016, the second CWI Shared task[3] was organized at Building Educational Applications workshop 2018[4]. This time the data was provided on four languages: English, German, Spanish and French. Whereas, for French, only the test set was available and no French training data. English corpora were extended and involved three genres: news, Wikinews and Wikipedia data. For comparison, on SemEval-2016 the corpora were formed from only Simple Wikipedia data. In 2018 the aim of the CWI Shared task was to identify words that are challenging for non-native speakers based on the annotations collected from both native and non-native speakers. The analysis (Yimam et al., 2018) of 12 submitted systems and 11 system description papers from 30 teams shows that traditional feature engineering-based approaches (mostly involving word length and frequency features) still perform better than neural network and word embedding-based approaches. This time much more participants used deep learning approach in their solutions, which resulted in significant improvement of performance in CWI task on

[2] https://nlp.stanford.edu/projects/glove/

[3] https://sites.google.com/view/cwisharedtask2018/

[4] http://www.cs.rochester.edu/~tetreaul/naacl-bea13.html

monolingual English track: the top rank systems reached from 81.1 to 87.4 F-score for different English datasets. At the same time cross-lingual German, Spanish and French tracks resulted in slightly lower F-score: 74.5, 76.9 and 75.9 correspondingly. Nevertheless, cross-lingual results were considered highly promising. This point was the most important finding of this shared task.

Among the deep learning solutions used for resolving the CWI 2018 Shared Task there were:

- application of Convolutional Neural Network (CNN) for the first time for CWI task (Aroyehun et al., 2018). The solution is based on 2D convolution and word-embedding representation of the target text fragment and its context. The CNN-based system did not show significant improvement in performance compared to an alternative system based in feature engineering and Tree Ensembles developed by the same team.

- a DNN which was feed with both word-level and character-level embeddings (De Hertog and Tack, 2018). The word-level representations were trained by team on their own on COW-corpora[5] with gensim[6] implementation of word2vec model (described in the next chapter 3.4.1). The character-level embeddings were trained by the DNN itself when learning to classify words into complex and non-complex.

In contrast to the last solution, in this work, we test the performance of FastText (described in the next chapter 3.4.2), which is a word2vec's modification and captures not only distributional properties of words but also morphological ones, as this model is trained on subword instead of word level. And again, in CWI 2018 Shared Task words were provided in context, which is different to our setting.

2.3.2 Medical area

Not so much effort has been devoted to the exploitation of NLP potential in the measurement of readability of medical texts. In the biomedical domain, the readability assessment currently is approached as a classification task as well as in general language. The difference is that here a much smaller variety of features has been tested. The following feature types are mostly used for processing of biomedical documents:

- a combination of classical readability formulas with medical terminologies (Kokkinakis and Toporowska Gronostaj, 2006);

- n-grams of characters (Poprat, Markó, and Hahn, 2006);

- stylistic (Grabar, Krivine, and Jaulent, 2007) or discursive (Goeuriot, Grabar, and Daille, 2008) features which characterize the discourse of documents;

- lexicon features, for example, lexical density - the number of unique number of words within a given unit (e.g. sentence, document) (Miller et al., 2007);

- morphological features (Chmielik and Grabar, 2011);

- combinations of different features from the listed above (Zeng-Treiler et al., 2007).

[5] `https://corporafromtheweb.org/`

[6] `https://radimrehurek.com/gensim/`

Among the recent experiments dedicated to readability study in the medical domain are, for example, the following:

- manual rating of medical words (Zheng, Milios, and Watters, 2002),
- automatic rating of medical words on the basis of their presence in different vocabularies (Borst et al., 2008),
- exploitation of machine learning approach with various features (Grabar, Hamon, and Amiot, 2014).

The last experiment achieved up to 85.0 F-score on individual annotations.

Due to the recent significant advance in the study of readability in general language and the relatively slow progress in the medical area, there is a great potential to experiment with the application of various machine learning-based approaches on medical texts. This fact motivated us for this work.

Chapter 3

Background Information

In this chapter, we will cover in brief the basic notion regarding the methodology we propose for the detection of word difficulty. The methodology itself is described in chapter 5.

3.1 Classification problem

Classification is a supervised machine learning problem. Given a set of n attributes (features), a set of k classes and described by a set of m labeled training instances

$$\{(x_i, y_i); i = 1, ..., m\},$$

where x_i is a feature vector and y_i is a label, the task is to find such a model, which predicts the class of any instance from the values of its attributes.

A lot of real-world problems can be considered as a classification problem, for instance (Ng, 2012):

- understanding whether a tumor is malignant or benign by its size,
- distinguishing spam and non-spam emails by the words they contain,
- identifying fraudulent transactions among normal ones using their metadata.

To handle those tasks many classification algorithms currently exist, among which the most commonly used groups are linear classifiers, support vector machines, nearest neighbors classifiers, decision trees, artificial neural networks.

We will concentrate on the last two further in this chapter.

3.1.1 Classifier performance evaluation

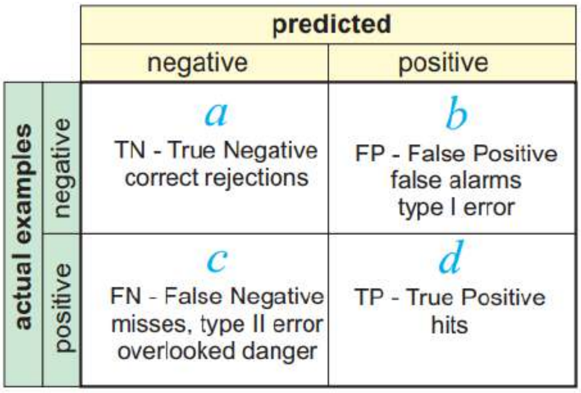
When training any machine learning model, the full set of available data is commonly split into several parts:

1. Training set - the sample of data used for training (fitting) a model. This is the only set with target variable (labels in case of classification) "visible" for a model. In the case of all the rest of datasets, the target variable is only used for performance evaluation of the fitted model.
2. Validation or development set - the sample of data used for unbiased evaluation of a model fit on training set while tuning the model's hyperparameters. In other words, this set is needed to choose a model which will be finally used in production.

3. Test set - the sample of data used for unbiased evaluation of the final model, which was fitted on training dataset (Kuhn and Johnson, 2014).

To evaluate a classification model predicted labels results are compared with class labels provided in the development or the test set. This allows checking of the generalization ability of the model.

For the simplicity of the explanation of how a classifier is evaluated, we will consider the evaluation of a binary classifier, which has only two target classes for prediction: positive and negative. Binary classifiers are mostly evaluated using a confusion matrix (fig. 3.1).



		predicted	
		negative	positive
actual examples	negative	*a* TN - True Negative correct rejections	*b* FP - False Positive false alarms type I error
	positive	*c* FN - False Negative misses, type II error overlooked danger	*d* TP - True Positive hits

FIGURE 3.1: Confusion matrix. Source: (Kohavi, 1998).

Performance measures calculated from the confusion matrix entries are the following (Sebastiani, 2002):

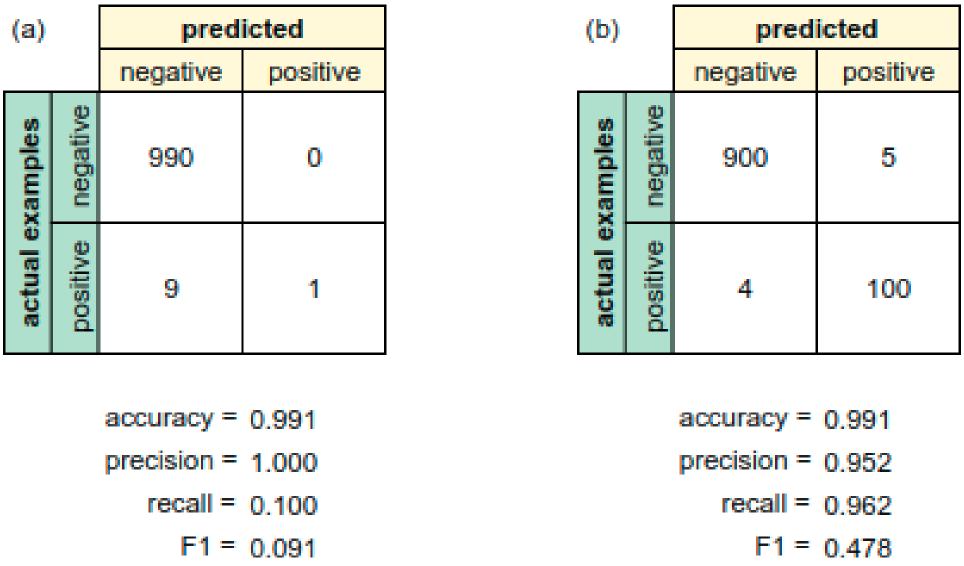
- Accuracy $= (a+d)/(a+b+c+d) = (TN+TP)/total$;
- True positive rate, recall, sensitivity$= d/(c+d) = TP/actual\ positive$;
- Specificity, true negative rate $= a/(a+b) = TN/actual\ negative$;
- Precision, predicted positive value $= d/(b+d) = TP/predicted\ positive$;
- False positive rate, false alarm $= b/(a+b) = FP/actual\ negative = 1 - specificity$;
- False negative rate $= c/(c+d) = FN/actual\ positive$.

One of the measures above is not enough to evaluate a binary classifier properly when data is class imbalanced. For instance, in fig. 3.2 the accuracy is high and equal for both situations, but precision and recall differ significantly.

Moreover, it is always a question what to prioritize, precision or recall, and how to find balance among these two measures. For this reason, the F_1 score, a harmonic mean of precision and recall (Chinchor, 1992), is frequently used to evaluate a binary classifier:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.1}$$

In our experiments described in chapter 6 we work with multiclass classification problem on unbalanced datasets (Table 4.1). The quality of the applied classification



(a)

		predicted	
		negative	positive
actual examples	negative	990	0
	positive	9	1

accuracy = 0.991
precision = 1.000
recall = 0.100
F1 = 0.091

(b)

		predicted	
		negative	positive
actual examples	negative	900	5
	positive	4	100

accuracy = 0.991
precision = 0.952
recall = 0.962
F1 = 0.478

FIGURE 3.2: Examples of a classifier evaluation.

algorithms is evaluated using four standard measures: accuracy A, precision P, recall R and F1-measure F. To effectively measure the ability of a model to distinguish between three target classes of words in an unbalanced dataset we use *macroaveraging* of three one-vs-rest binary classifiers. Macroaveraging is a method to measure multiclass classifier in case of unbalanced dataset (Sebastiani, 2002). Precision and recall are first evaluated 'locally' for each class and then 'globally' by averaging over the results of the different categories:

$$P^M = \frac{\sum_{i=1}^{|C|} P_i}{|C|},\ R^M = \frac{\sum_{i=1}^{|C|} R_i}{|C|}, \tag{3.2}$$

where:

M = for macroaveraging,
$C = \{c_1, ..., c_{|C|}\}$ - set of classes,
$|.|$ = capacity (number of samples),
$|C|$ = total number of samples in the dataset.

3.1.2 Cross-validation

Cross-validation (also called *out-of-sample testing* or *rotation estimation*) is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set (Kohavi, 1995). To perform a cross-validation the full dataset D is randomly split into k mutually exclusive subsets (the **folds**) $D_1, D_2, ..., D_3$ of approximately equal size. Then the model is trained and tested k times. Each time $t \in \{1, 2, ..., k\}$ it is trained on all fords except for t^{th} one: D_t, - and tested on D_t. The chosen performance measure π is calculated at each time t on test set and then averaged resulting to cross-validation performance estimate: $\pi_{CV} = 1/n \sum_{t=1}^{k} \pi_i$.

Mostly cross-validation is performed within the instances of a dataset (rows in case of table data). Whereas in this work we propose to cross-validate also within different target columns (annotations) and within instances and target columns simultaneously (chapter 5).

3.2 Decision trees

Decision trees can be used for both regression and classification tasks. Learning a decision tree is the construction of a tree-like model out of class-labeled training tuples. An example of a decision tree is shown on fig. 3.3. Such model is, in fact, a sequence of conditional control statements based on values of feature vectors characterizing input observations.

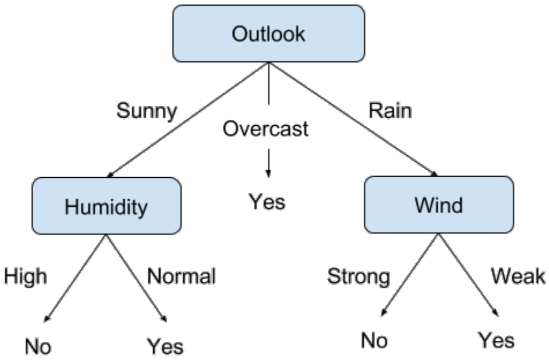


FIGURE 3.3: A decision tree for decision making about playing tennis.

A decision tree is mostly learned using the recursive binary splitting technique. In this process, at each step an algorithm is aimed to find the best feature and splitting condition to finally come up with the shortest path to the final decision. Learning an optimal binary decision tree in such a way is an NP-complete problem (Hyafil and Rivest, 1976). For this reason, on practice greedy heuristic algorithms are used, where locally optimal decisions are made at each node. In this work we use an implementation of the popular ID3 (Iterative Dichotomiser 3) algorithm (Quinlan, 1986) in all our experiments 6. Using decision trees in applications of evidence-based medicine is common as this model is conceptually simple, provides high accuracy and mimics the way a doctor thinks (Sackett et al., 1996; Podgorelec et al., 2002). Moreover, decision trees give a clear explanation of the class choice, which is valuable for medicine where it is important to have the traceability of the results. For NLP tasks, decision trees similarly, provide valuable information on the relevant features and/or feature values.

To sum up, the advantages of decision trees are:

- simplicity of concept and interpretability,
- possibility to apply to both categorical and numeric data without the need for regularization,
- easiness to combine with other decision techniques.

Disadvantages of decision trees are:

- instability - a small change in data can lead to a dramatic change in the model,
- propensity of overfitting if no constraints are put. To avoid this effect in our experiments, we restricted the depth of trees as specified in tables of chapter 6.

3.3 Artificial neural networks

An Artificial Neural Network (ANN) is a mathematical model which consists of interconnected layers of neurons (groups of nodes) as shown on fig. 3.4. Layers can be of different types, and it is common to stack several distinct layers together in a specific manner to get a neural network with good performance. The model in fig. 3.4 contains two *fully-connected* hidden layers where neurons are fully pairwise connected between two adjacent layers (Li, Karpathy, and Johnson, 2016). Artificial neural networks that contain multiple hidden layers are called Deep Neural Networks (DNN).

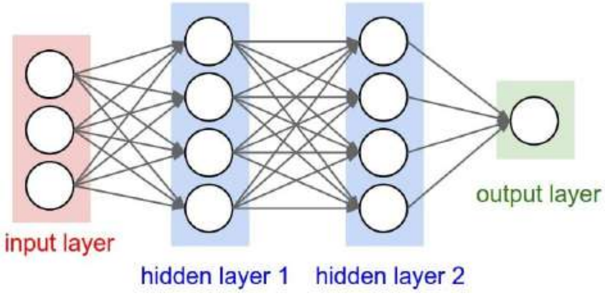


FIGURE 3.4: A 3-layer neural network with three inputs, two hidden layers of 4 neurons each and one output layer. Source: (Li, Karpathy, and Johnson, 2016)

A neural network can be used for both supervised and unsupervised tasks. In the case of supervised training (such as classification), the network processes inputs and provides outputs (predictions). Predictions are then compared with the correct values of target variable values. Errors are then propagated back through the system. This process results in adjusted weights of a neural network. The aim is after many iterations of the described procedure to receive well-tweaked weights of the neural network which provide satisfactory performance of the model.

As ANNs contain a lot of connections (which are expressed as weights in mathematical model), much data needs to pass through the model to train it well. As the data passed to model is stored in random-access memory (RAM) of the working machine during forward and backward passes, we mostly cannot pass all the available data at once to our model. That is why data is split by small portions called *batches*. A complete pass of a given dataset through the model is called *epoch*. It is one iteration of learning. A different number of epochs is needed to train different DNN architectures, from 5 to hundreds. In this work, we trained neural networks for at most 16 epochs, which was enough for our task.

3.3.1 Recurrent neural networks

A Recurrent Neural Network (RNN) is a type of ANN for handling sequential data by processing them element-wise and storing in the internal memory (hidden state). A part of an RNN is presented on fig. 3.5.

Formally, forward propagation of an RNN begins with initialization of the initial hidden state $h^{(0)}$ and then for each time step from $t = 1$ to $t = \tau$ the following update equations are applied (Goodfellow, Bengio, and Courville, 2016):

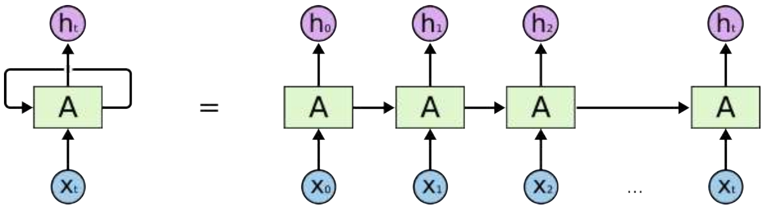


FIGURE 3.5: A rolled (on the left) and unrolled (on the right) part of vanilla (or conventional) RNN. At each time step t the model takes the t^{th} member of sequence x_t and outputs a hidden state value h_t. Source: (Olah, 2015)

$$\begin{aligned} a^{(t)} &= b + Wh^{(t-1)} + Ux^t, \\ h^{(t)} &= tanh(a^{(t)}), \\ o^{(t)} &= c + Vh^{(t)}, \\ y^{(t)} &= softmax(o^{(t)}), \end{aligned} \tag{3.3}$$

where x^t is an input vector, y^t is an output vector and the parameters are weight matrices U, V, W - for input-to-hidden, hidden-to-output and hidden-to-hidden connections respectively, and bias vectors b and c.

Due to the specifics of RNNs, these models are widely used not only for solving classical NLP ones like machine translation (Chen et al., 2018) and part-of-speech tagging (Plank, Søgaard, and Goldberg, 2016), but also for handling sequence-based tasks from other domains like program code generation (Stehnii, 2017) and filling missing values in multivariate time series (Che et al., 2016).

Long short-term memory units

As it follows from fig. 3.5, the vanilla RNN processes information sequentially through time in both directions, forward and backward. This means that the signal can be easily corrupted when multiplied on small numbers (near 0) several times. This is known as the *vanishing gradients* problem, which happens during backpropagation. This prevents a vanilla RNN from memorizing long-term dependencies.

Long short-term memory (LSTM) units are designed with the idea of getting rid of this problem. In LSTM, the repeating module has a different structure than vanilla RNN (fig. 3.6). Instead of having a *single* neural network layer, there are *four* of them, interacting in a very special way. The key *distinctive feature* of LSTM is the top horizontal line which runs down the entire chain with minor linear interactions. So the information flows with this part with almost no changes (Olah, 2015).

The vanilla LSTM and its modifications are frequently used for building sequence-processing systems and show an advantage in performance compared to other RNN units (Hochreiter and Schmidhuber, 1997). In this work, after experimenting with different RNN architectures, an LSTM-based one showed the best results for our task A.

3.4 Vector words representations

Word embedding is a collective name for a set of language modeling and feature learning techniques in NLP where words or phrases from vocabulary are mapped to real-valued vectors in a predefined low-dimensional space. This concept is widely

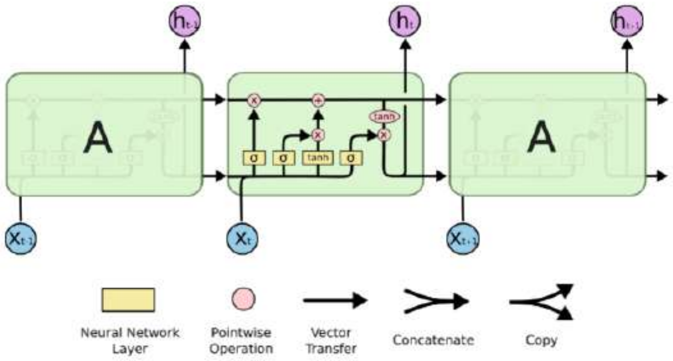


FIGURE 3.6: The repeating module in an LSTM contains four interacting layers. Below are explanations of symbols used on the diagram. Source: (Olah, 2015)

used nowadays in applications of NLP as working with word vectors of vocabulary size (thousands or millions of dimensions) is computationally hard. Whereas each high dimensional vector is sparse containing only a few "valuable" non-zero values, which led to the idea that words can be represented much more densely (Harris, 1954). Mathematically this process involves a mathematical embedding from space with one dimension per word to a continuous vector space with a much lower dimension (mostly from 100 to 300 dimensions) (Brownlee, 2017).

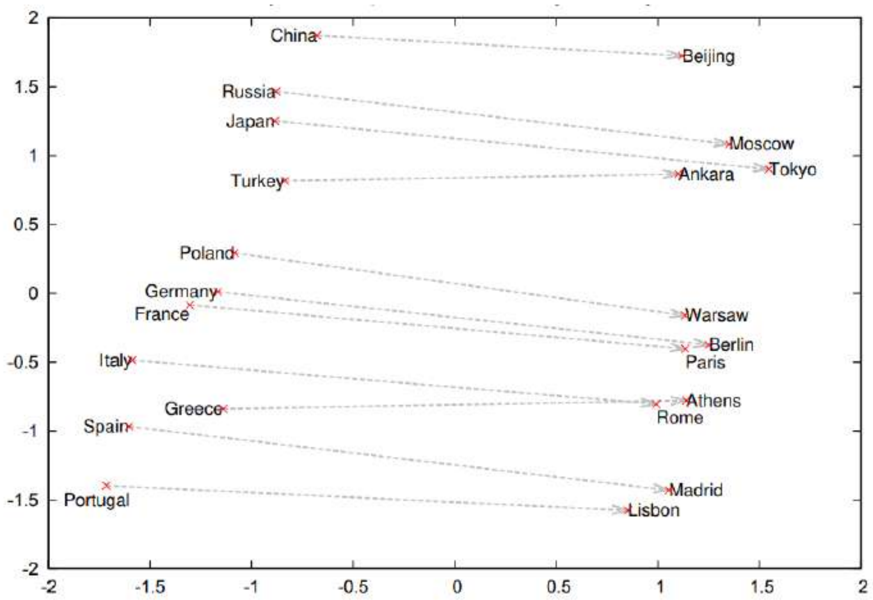


FIGURE 3.7: Country and Capital Vectors Projected by PCA (Principal Component Analysis). The figure illustrates the ability of the word2vec model to organize concepts and learn the relationships between them implicitly. No supervised information about country-capital correspondence was provided to the model during learning. Source: (Mikolov et al., 2013a)

3.4.1 Word2vec

Word2vec is a well-known deep-learning-based approach for receiving word embeddings which has seen tremendous success being applied in numerous NLP tasks due to its computationally-efficiency and high quality of result Mikolov et al., 2013a. Word2vec representations can be trained using either skip-gram model (Mikolov et al., 2013a) shown on fig. 3.8 or Continuous Bag-of-Words model (CBOW) (Mikolov et al., 2013b) shown on fig. 3.9.

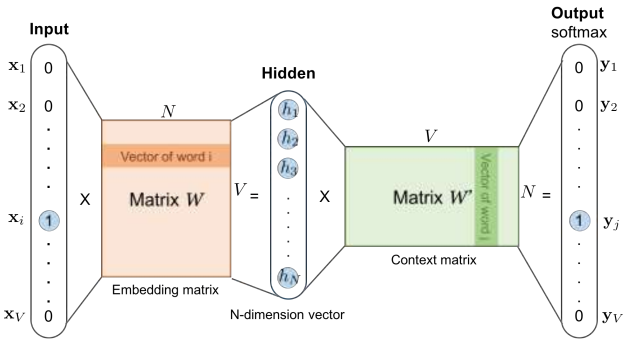


FIGURE 3.8: The skip-gram model. Both the input vector x and the output y are one-hot encoded word representations. The hidden layer is the word embedding of size N. Source: (Weng, 2017)

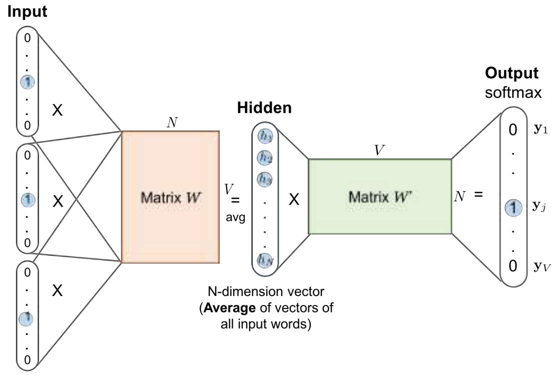


FIGURE 3.9: The CBOW model. Word vectors of multiple context words are averaged to get a fixed-length vector as in the hidden layer. Source: (Weng, 2017)

A nice property of word2vec word representation is that due to the way word embeddings are learned, final vectors capture context information of words, which results in the existence of semantic word relationships, an example of which is shown on fig. 3.7.

For this reason, such vectors are frequently used as features for many canonical NLP prediction tasks, such as part-of-speech tagging, named entity recognition (Collobert et al., 2011), or classification. In our work for the same purpose, we utilized fastText word embeddings - an advanced modification of word2vec model.

3.4.2 FastText word representations

In NLP applications it is common to use pre-trained word representations estimated from a large corpus of non-domain-specific texts: news collections, Wikipedia, Web Crawl. In a vanilla word2vec setting, word embeddings map each word to a distinct vector ignoring morphology. Moreover, words with low frequency in training corpora might be excluded from the final set of word-vector correspondences. This leads to a need of handling the problem of out-of-vocabulary (OOV) words when applying pre-trained word representations to a new corpus. When the domain of new texts contains much specific terminology, as in the case of our dataset of medical terms described in Chapter 4, the OOV problem becomes tough to resolve.

FastText is a skip-gram model trained on *n-grams* level introduced in Bojanowski et al., 2017. In this approach, each word is represented as a sum of character n-gram representations of this word's components. This makes it easy to represent any OOV word. Pre-trained word vectors for 157 languages using fastText were received and made public [1] by Facebook AI Research in 2017 (Mikolov et al., 2017). The results of applying those vectors for the French language on our dataset are described in 6.2.

FastText embedding vectors are the sum of character n-gram representations so that they could be generated even for unknown words.

In this chapter, we provided the background information which briefly describes all the components of our proposed methods. In the next chapter, we will describe the dataset used in experiments.

[1] `https://fasttext.cc/docs/en/crawl-vectors.html`

Chapter 4

Dataset description

For the experiments with the supervised word categorization task, we used the publicly available set of words with annotations[1] collected according to the procedure described in the source work (Grabar and Hamon, 2016). Additionally, for the research of generalization abilities of our models described in 6.3, we were provided with four more sets of annotations. The process of word collection and annotation is briefly described below.

4.1 Linguistic data description

The set of required biomedical terms was obtained from the French part of Snomed International (Côté et al., 1993) - a medical terminology, available from the ASIP SANTE website[2]. The purpose of the terminology stored here is to provide an extensive up-to-date overview of the medical field. Snomed contains 151,104 medical terms organized into eleven semantic axes such as disorders and abnormalities, procedures, chemical products, living organisms, anatomy, and social status. For word understandability study, five axes relating to the main medical notions were chosen: disorders, abnormalities, procedures, functions, and anatomy. These categories are assumed to contain terms which are familiar to a layman, in contrast to contents of such specific groups as chemical products (*hydrogen sulfide*) and living organisms (*Sapromyces*, *Acholeplasma laidlawii*).

The 104,649 selected terms were then processed. First, they were tokenized into words (or tokens) using TreeTagger (Schmid, 1994). Then the result was lemmatized with FLEMM - a lemmatizer for French texts (Namer, 2000). After that we received 29,641 unique words, for instance, the term '*trisulfure d'hydrogène*' provided three words (*trisulfure*, *de*, *hydrogène*).

The final dataset contains three morphological groups of words:

- compound words which contain several bases: abdominoplastie (abdominoplasty), dermabrasion (dermabrasion);
- constructed words which contain one base and at least one affix: cardiaque (cardiac), acineux (acinic), lipoïde (lipoid);
- simple words which contain one base, no affixes and possibly infections (when the lemmatization fails): acné (acne), fragment (fragment).

[1] http://natalia.grabar.free.fr/resources.php#rated

[2] http://esante.gouv.fr/services/referentiels/referentiels-d-interoperabilite/snomed-35vf

4.2 Annotation process

The set of 29,641 unique words was annotated by seven French speakers, 25-40-year-old, without medical training, without specific medical problems, but with a linguistic background. The annotators were expected to represent the average knowledge of medical words among the population as a whole. The annotators were presented with a list of terms and asked to assign each word to one of the three categories:

- I can understand the word;
- I am not sure about the meaning of the word;
- I cannot understand the word.

The assumption is that the words, which are not understandable by the annotators, are also difficult to understand for the patients. The annotators were asked not to use dictionaries during the annotation process. The annotation results are represented in Table 4.1.

Annotators / Categories	*1. I can understand*	*2. I am not sure*	*3. I cannot understand*	*Total annotations*
O1 (%)	8,099 (28)	1,895 (6)	19,647 (66)	29,641
O2 (%)	8,625 (29)	1,062 (4)	19,954 (67)	29,641
O3 (%)	7,529 (25)	1,431 (5)	20,681 (70)	29,641
A1 (%)	11,680 (39)	2,312 (8)	15,649 (53)	29,641
A2 (%)	9,108 (31)	2,994 (10)	17,539 (59)	29,641
A7 (%)	10,606 (36)	2,206 (7)	16,829 (57)	29,641
A8 (%)	7,735 (26)	1,032 (3)	20,874 (70)	29,641

TABLE 4.1: Number (and percentage) of words assigned to reference categories by seven annotators (O1, O2, O3, A1, A2, A7, A8).

In this chapter, we introduced our data source which consists of 29,641 unique French preprocessed words annotated by seven French speakers who are non-specialists in the medical sphere. The annotation characterizes the understandability of the word for annotator. As a result, each word was assigned one of three categories. In the next chapter, we will describe the process of feature generation for the words from our dataset and resolving a multiclass classification task based on user annotations.

Chapter 5

Methodology

The purpose of this work is to categorize medical words according to whether they can be understood or not by non-specialized people, using features obtained with deep learning methods. The manual annotations of these words described in the previous chapter provide the reference data. The proposed method includes three steps:

1. calculation of NLP features associated with the annotated words;
2. training a machine learning model for word classification;
3. evaluation of classification quality using cross-validation.

In this research we want to provide answers to the following questions:

1. Which feature set distinguishes better between understandable and non-understandable medical words?
2. Why one feature set categorizes better than another?
3. Do classifiers built on the considered feature sets generalize well?

5.1 Feature sets

5.1.1 Standard NLP features

We will refer to the previously used NLP features (Grabar, Hamon, and Amiot, 2014) as *"standard features"* opposed to two kinds of *"embeddings"* described in the next subsection. The standard features include 24 linguistic and extra-linguistic features related to general and specialized languages. The features are computed automatically and can be grouped into ten classes:

- *Syntactic categories.* Syntactic categories and lemmas are computed by TreeTagger (Schmid, 1994) and then checked by FLEMM (Namer, 2000). The syntactic categories are assigned to words within the context of their terms. If a given word receives more than one category, the most frequent one is kept as a feature. Among the main categories, we find for instance nouns, adjectives, proper names, verbs, and abbreviations.
- *Presence of words in reference lexica.* Two reference lexica of the French language were used: TLFi[1] and *lexique.org*[2]. TLFi is a dictionary of the French language covering XIX and XX centuries. It contains almost 100,000 entries. *lexique.org*

[1] http://www.atilf.fr/

[2] http://www.lexique.org/

is a lexicon created for psycholinguistic experiments. It contains over 135,000 entries, among which inflectional forms of verbs, adjectives, and nouns. It contains almost 35,000 lemmas.

- *Frequency of words through a non-specialized search engine.* For each word, a query to Google search engine was sent in order to find out the frequency of the word attested on the web.
- *Frequency of words in the medical terminology.* The frequency of words in the medical terminology Snomed International was computed.
- *Number and types of semantic categories associated with words.* The information on the semantic categories of Snomed International was used.
- *Length of words in a number of their characters and syllables.* For each word, the number of its characters and syllables was computed.
- *Number of bases and affixes.* Each lemma was analyzed by the morphological analyzer Dérif (Namer and Zweigenbaum, 2004), adapted to the treatment of medical words. It performs the decomposition of lemmas into bases and affixes known in its database, and it also provides a semantic explanation of the analyzed lexemes. The morphological decomposition information (number of affixes and bases) was exploited.
- *Initial and final substrings of the words.* Initial and final substrings of different length, from three to five characters, were computed.
- *Number and percentage of consonants, vowels and other characters.* The number and the percentage of consonants, vowels and other characters (i.e., hyphen, apostrophe, comas) was computed.
- *Classical readability scores.* Two classical readability measures were applied: Flesch (Flesch, 1948) and its variant Flesch-Kincaid (Kincaid et al., 1975). Such measures are typically used for evaluating the difficulty level of a text. They exploit surface characteristics of words (number of characters and/or syllables) and normalize these values with specifically designed coefficients.

5.1.2 FastText word embeddings usage

FastText word embeddings (described in section 3.4.2) are a good choice for getting word features in difficulty detection task because they are able to use words' morphological information and generalize over it. The fact that word embeddings capture context and morphological information leads to the hypothesis that incorporating this information as features will improve classification accuracy for our specific problem.

We found out that FastText word embeddings trained on Wikipedia and Common Crawl[3] texts have a quite large portion of *known* (learned) words from our dataset. According to our analysis, 44.26% (13,118 out of 29,641) medical words in the dataset and 56.00% (16,598 out of 29,641) lowercased medical words in the dataset were used for training of the currently published FastText[4] model for French.

[3] http://commoncrawl.org/

[4] https://fasttext.cc

5.1.3 French RNN Medical Understandability Text Embeddings (FrnnMUTE)

According to the general functionality of RNN expressed in 3.3.1, the final hidden state aggregates the information about the whole input sequence. This idea is frequently used to receive hidden representations of sequences. Sequence-to-sequence (seq2seq) models are a well-known example of how this idea works in practice (Sutskever, Vinyals, and Le, 2014). Such models consist of two parts: an *encoder* is an RNN which encodes input sequence into a representation in hidden space (which is also called *thought vector*), and a *decoder* which generates a new sequence out of the hidden representations (fig. 5.1).

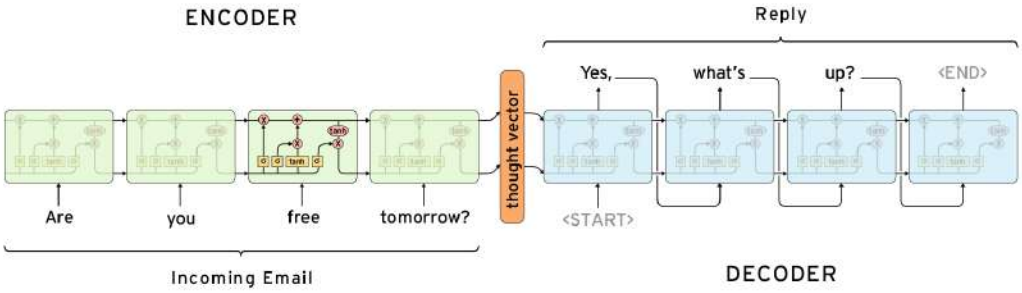


FIGURE 5.1: A seq2seq model for question answering task. Source: (Britz, 2016)

We utilized this idea for representing words from our dataset. To receive word representations from an RNN, we first trained it to classify words based on labels by one annotator (we chose $O1$), then for each word we found values of the last hidden state of the RNN and used this vector as features in word understandability detection for different users.

As a direct classifier, we trained a character-level RNN using PyTorch framework[5] and one GPU Tesla K80. For training we lowercased all words, converted them to a singular form and substituted all Unicode symbols with ASCII analogs. We tried several RNN architectures and hyperparameter sets; the detailed information is available in Appendix A.

We got the best F score macroaveraged (sec. 3.1.1) on three classes for the RNN with two unidirectional long short-term memory (LSTM) units (described in 3.3.1), each with 50 hidden units. The dropout of the model is 0.7. The input size is 57 as the number of unique characters in lowercase and converted to ASCII input words. The output size is 3 as this is the number of classes in our data.

This model reached the best performance on the eighth epoch with $F1 = 78.94$ and $accuracy = 81.21\%$ on development set. Using this model we received 50-dimensional word representations which we called FrnnMUTE (French RNN Medical Understandability Text Embeddings).

5.2 Cross-validation scenarios

For a thorough study of generalization abilities of the developed in this work classification models, we propose to consider three distinct cross-validation scenarios

[5] https://pytorch.org/

based on different combinations of users and vocabulary in train and test sets (fig. 5.2).

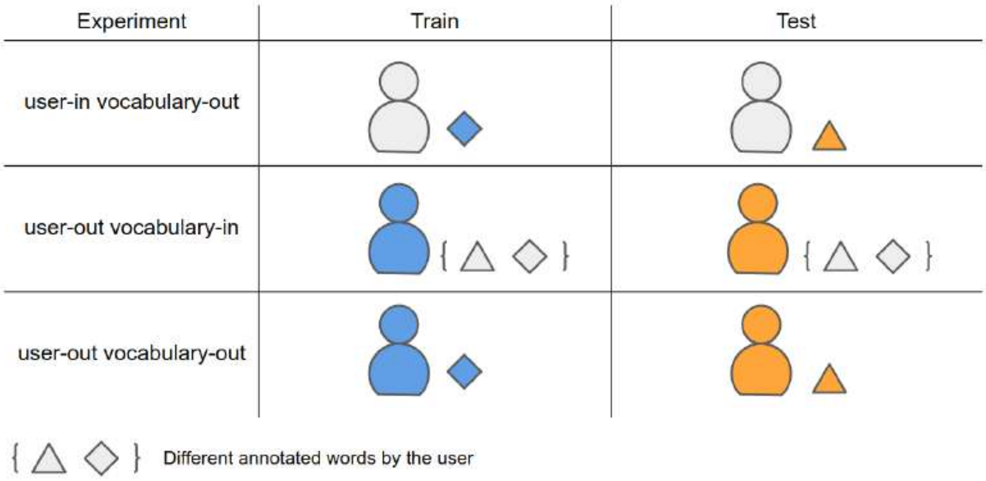


FIGURE 5.2: Visual description of the cross-validation experiments. There are experiment types by rows. By columns, there are combinations of a user (annotator) and a set of vocabulary used during training and testing. A user is depicted as a human icon. Different colors of human icons in columns mean that different users were used on training and testing stages. Different shapes of geometrical figures depict subsets of vocabulary. The logic for colors is the same as for human icons.

1. **User-in vocabulary-out cross-validation.** This type of experiment follows the scenario from the paper that we are comparing the results with throughout this work (Grabar, Hamon, and Amiot, 2014). The cross-validation is done on each dataset (i.e., each user's annotation) separately. The goal of these experiments is to measure the ability of the method (classification model) to generalize class recognition on the *known user* and his known manner to annotate words (that is, his understanding of the meaning of medical words) for *unknown words*. From the practical perspective, *user-in* means learning the profile of a user. So a model trained by such scenario represents the words understanding or knowledge of the annotator.

2. **User-out vocabulary-in cross-validation.** In this experiment, we learn from all the annotations of one user and then test the model on annotations of another user. Thereby, in such a setting, we measure the ability of the classifier to generalize on all known words, but for unknown users. This scenario is realistic to a real-world situation: the reference annotations can be obtained only from a couple of users, presumably representing the overall population, but not from all the possible users. Yet, it is necessary to predict the familiarity of medical words for all the potential users even if they did not participate in the annotations. In this scenario, the model learns the profile of a user, and we want to identify whether a new user has the same profile as an another. If the model predicts well for a new user, then it can be used for the identification of incomprehensible words for the new user.

3. **User-out vocabulary-out cross-validation.** In this experiment, we use (k-1) folds of data annotated by one user for training and test on the k-th fold of

data with annotations by the other user. In this case, we measure the ability of the method to generalize both on *unknown users* and *unknown vocabulary*. This experiment should be helpful in identifying the number of words needed for determining whether the profile of one user is the same as another in case the model shows good performance.

In this chapter, we introduced the methods which are tested in the experiments in the next chapter. Concretely, we explain our idea of using pre-trained FastText word embeddings for the detection of word difficulty. Also, we describe the process of receiving the novel FrnnMUTE embeddings. Finally, we introduce the three cross-validation scenarios which we will consider during experiments and which go beyond the standard cross-validation described in section 3.1.2.

Chapter 6

Experiments

We conducted a series of experiments to study the impact of adding vector word representations as features for a classification model on the quality of the word categorization. As in this work we compare results with ones in Grabar, Hamon, and Amiot, 2014, we first, reproduce results from the paper on the same datasets. Then we check how FastText word embeddings influence the quality of classification in different cross-validation scenarios. We notice that in one scenario FastText word embeddings significantly and confidently improve the performance of the classification model, so the next step is to study whether this model generalizes well on a greater variety of users. Finally, we study how FrnnMUTE used as features impact on classification quality in all the same cross-validation scenarios as considered previously and on all available user annotations.

6.1 Reproduction of previous results

In Grabar, Hamon, and Amiot, 2014 the classification methods were obtained using WEKA[1] - a collection of machine learning algorithms for data mining tasks implemented on Java. In our research as a tool to conduct experiments, we used Python as there are a lot of stable third-party Python libraries that make it convenient for research. In order to ensure the consistency of experiments in this work and in Grabar, Hamon, and Amiot, 2014, firstly, we reproduced the results in WEKA using the precomputed set of standard features described in the previous section 5.1.1 and *J48* classification algorithm - a WEKA implementation of C4.5 decision tree based algorithm described in Quinlan, 1993. Our results perfectly match with ones presented in the paper.

Secondly, we developed a solution in Python based on DT classifier from well-known scikit-learn library[2]. At this step we got 0.85-1.41 lower F scores for scikit-learn classifier compared to WEKA results (Table 6.1).

Since the input features were identical for both of WEKA and scikit-learn frameworks, we concluded that the little degradation of quality in case of using scikit-learn is caused by the difference in implementations of decision tree classifiers in these frameworks. In all subsequent experiments, we will use a scikit-learn classification DT model for the convenient comparison of experimental results. We will introduce slight changes in the depth of a DT for different dimensions of feature sets.

[1] https://www.cs.waikato.ac.nz/ml/weka/

[2] http://scikit-learn.org

user \method	*Results from paper (Grabar, Hamon, and Amiot, 2014)*	*WEKA J48*	*Python Decision trees (10-fold CV, with shuffle)*
*O*1	80.6	80.5	79.8
*O*2	81.4	80.9	80.0
*O*3	84.5	84.5	83.2

TABLE 6.1: F1 score. Comparison of different implementations of a decision tree classifier on three sets of annotations (O1, O2, O3) in user-in vocabulary-out cross-validation. The DT in scikit-learn was restricted to a depth not more than 3 (this showed the best result during grid-search of hyperparameters of the DT).

6.2 Experiments with cross-validation scenarios

6.2.1 User-in vocabulary-out cross-validation

We carried out the experiments using (i) the standard features only, (ii) the FastText word embeddings only and (iii) their combination. Experiments with isolated FastText word embeddings as features and the data from three annotators resulted in poor F1 scores (Table 6.2), that can be explained by the fact that contextual information which is dominant in the word embeddings is not enough to define word understandability. Adding the FastText word embeddings to the standard feature set resulted in up to a 1.0 higher F1 score due to higher Precision (up to 1.8), meaning that contextual information slightly impacts on the understandability of a word by a given person.

Train user	*Test user*	*Standard features*				*FastText embeddings*				*Standard features + FastText embeddings*			
		A	*P*	*R*	*F*	*A*	*P*	*R*	*F*	*A*	*P*	*R*	*F*
*O*1	*O*1	**82.5**	77.2	**82.5**	79.8	72.5	67	72.5	69.3	82.4	**79**	82.4	**80.2**
*O*2	*O*2	**82**	78.9	**82**	80	73.5	69.9	73.5	71.3	81.9	**79.5**	81.9	**80.3**
*O*3	*O*3	85.5	81.2	85.5	83.2	74.9	70.4	74.9	72.3	**85.9**	**83**	**85.9**	**84.2**

TABLE 6.2: F1 score. Experiments on user-in vocabulary-out cross-validation. The best score for a combination of quality measure and experiment among three feature sets is in bold.

6.2.2 User-out vocabulary-in cross-validation

In these experiments, we got a substantial improvement of combined features in comparison to the standard features (Table 6.3). When knowledge of word understandability of one user is used to predict it for another user, adding the FastText word embeddings provides up to 2.9 better F1 score. Notice that used separately, standard features and embeddings show similar performance as in user-in vocabulary-out cross-validation (Table 6.2). Our hypothesis is that there exists a robust nonlinear dependency between some subsets of standard features and subword-level components of FastText word embeddings. Testing this hypothesis is the topic of our further research.

Train user	*Test user*	*Standard features*				*FastText embeddings only*				*Standard features + FastText embeddings*			
		A	*P*	*R*	*F*	*A*	*P*	*R*	*F*	*A*	*P*	*R*	*F*
O1	*O2*	81.7	78.6	81.7	80.1	74	70.3	74	71.2	**84.2**	**82**	**84.2**	**82.8**
O1	*O3*	85	81.2	85	83	75.4	70.7	75.4	72.6	**87.6**	**84.9**	**87.6**	**85.9**
O2	*O1*	82.2	77	82.2	79.1	72.8	67.3	72.8	69.6	**83.9**	**80.2**	**83.9**	**81.1**
O2	*O3*	85.4	81.1	85.4	83	75.3	71.1	75.3	73	**86.8**	**83.5**	**86.8**	**84.7**
O3	*O1*	82.8	77.4	82.8	79.7	72.7	67.1	72.7	69.4	**84.9**	**81.3**	**84.9**	**82.4**
O3	*O2*	82.2	79	82.2	80.2	74.1	70.4	74.1	71.6	**84.2**	**82.1**	**84.2**	**82.8**

TABLE 6.3: F1 score. Experiments on user-out vocabulary-in cross-validation.

6.2.3 User-out vocabulary-out cross-validation

The cross-validation setting is now the most strict and knowledge of words understandability of one user is used to predict whether another user will understand other medical words. In these experiments, FastText word embeddings provide approximately 0.5% higher F1 score in case of learning on users O1 and O3 (Table 6.4). When learning on user O2, embeddings decrease F by 0.5, which means that annotations and health literacy of user O2 are different from users O1 and O3. It seems that adding embeddings overfits the machine learning model to the dataset. As a result, tests on the other "kind of word understandability" and combined features are less successful compared to using standard features only for learning. This may be due to the lack of systematicity in annotations of O2. We will also encounter this issue when more annotators are involved in experiments in sections 6.3 and 6.4.

Train user	*Test user*	*Standard features*				*FastText embeddings*				*Standard features + FastText embeddings*			
		A	*P*	*R*	*F*	*A*	*P*	*R*	*F*	*A*	*P*	*R*	*F*
O1	*O2*	81.7	78.6	81.7	80.1	73.6	69.9	73.6	71.3	**81.8**	**79.8**	**81.8**	**80.6**
O1	*O3*	**85**	81.2	**85**	83	74.8	70.4	74.8	72.4	84.9	**82.2**	84.9	**83.4**
O2	*O1*	**82.2**	76.9	**82.2**	**79.1**	72.5	66.9	72.5	69.3	81.7	**77.5**	81.7	**79.1**
O2	*O3*	**85.3**	81	**85.3**	**83**	75.1	70.7	75.1	72.7	84.4	**81.3**	84.4	82.5
O3	*O2*	**82.7**	77.3	**82.7**	79.7	72.5	66.9	72.5	69.2	82.6	**78.9**	82.6	**80.2**
O3	*O3*	82.1	79	82.1	80.1	73.8	70.2	73.8	71.4	**82.2**	**80**	**82.2**	**80.7**

TABLE 6.4: F1 score. Experiments on user-out vocabulary-out cross-validation.

6.3 Generalizability study

In the previous experiments, we concentrated on making three annotators' data consistent with the research in paper Grabar, Hamon, and Amiot, 2014. To study better generalizability of models for word understandability detection, we included four more annotators in an experiment.

In this part, we concentrate on the user-out vocabulary-in cross-validation scenario as the most realistic one. Here, understanding of the quality of generalization is crucial for usage of the model in a real-world client-doctor relationship.

Train user	*Test user*	*Standard features*			*FastText embeddings*			*Standard features + FastText emb*		
		P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
*O*1	*O*1	77.2	82.5	79.7	67.0	72.5	69.3	79.0	82.4	80.2
*O*1	*O*2	78.6	81.7	80.1	70.3	74.0	71.2	82.0	84.2	82.8
*O*1	*O*3	81.2	85.0	83.0	70.7	75.4	72.6	84.9	87.6	85.9
*O*1	*A*1	71.0	74.7	71.2	62.1	63.8	58.8	74.1	75.4	72.2
*O*1	*A*2	70.6	78.4	74.0	61.9	68.5	63.3	75.0	80.1	76.2
*O*1	*A*7	72.6	77.5	74.2	63.0	66.6	61.9	76.2	78.9	75.8
*O*1	*A*8	82.3	84.9	83.5	73.1	76.8	74.5	85.7	87.8	86.6
*O*2	*O*1	77.0	82.2	79.1	67.3	72.8	69.6	80.2	83.9	81.1
*O*2	*O*2	78.9	82.0	80.0	69.9	73.5	71.3	79.5	81.9	80.3
*O*2	*O*3	81.1	85.4	83.0	71.1	75.3	73.0	83.5	86.8	84.7
*O*2	*A*1	71.1	72.1	68.2	61.7	64.5	60.2	74.0	75.1	71.5
*O*2	*A*2	70.8	77.3	72.7	61.8	68.9	64.2	76.0	79.8	75.5
*O*2	*A*7	72.7	75.6	71.8	62.6	67.0	62.8	75.9	78.3	74.9
*O*2	*A*8	83.0	86.2	84.4	73.7	77.1	75.3	85.4	88.2	86.7
*O*3	*O*1	77.4	82.8	79.7	67.1	72.7	69.4	81.3	84.9	82.4
*O*3	*O*2	79.0	82.2	80.2	70.4	74.1	71.6	82.1	84.2	82.8
*O*3	*O*3	81.2	85.5	83.2	70.4	74.9	72.3	83.0	85.9	84.2
*O*3	*A*1	71.8	73.3	69.5	61.7	64.1	59.6	75.1	75.4	72.1
*O*3	*A*2	71.2	78.0	73.5	61.8	68.7	63.9	76.8	80.2	76.3
*O*3	*A*7	73.2	76.5	72.9	62.4	66.6	62.2	77.2	78.8	75.8
*O*3	*A*8	82.6	85.8	84.1	73.7	77.2	75.2	86.0	88.0	86.9
*A*1	*O*1	77.2	82.5	79.8	66.5	67.9	66.6	76.9	79.5	77.6
*A*1	*O*2	78.6	81.6	80.1	69.2	69.0	68.5	78.8	79.6	78.9
*A*1	*O*3	81.2	84.9	82.9	70.7	69.6	69.2	81.8	82.0	81.0
*A*1	*A*1	70.9	74.7	71.3	59.4	64.6	61.8	72.4	75.1	72.9
*A*1	*A*2	70.5	78.3	74.0	60.6	66.4	63.2	73.7	78.6	75.0
*A*1	*A*7	72.6	77.5	74.2	61.3	66.1	63.6	75.1	79.2	76.5
*A*1	*A*8	82.2	84.8	83.5	72.3	70.4	70.4	81.5	81.0	80.5
*A*2	*O*1	77.3	82.6	79.8	67.2	72.6	69.6	81.0	82.8	81.8
*A*2	*O*2	78.6	81.6	80.1	70.4	74.0	71.9	82.0	82.0	82.0
*A*2	*O*3	81.2	84.9	83.0	71.0	75.2	73.0	84.9	85.4	85.1
*A*2	*A*1	70.9	74.6	71.2	61.5	64.6	60.4	76.5	76.5	74.7
*A*2	*A*2	70.6	78.4	74.0	61.2	68.4	63.7	74.7	77.8	75.6
*A*2	*A*7	72.6	77.5	74.2	62.4	67.0	63.0	77.6	78.9	77.3
*A*2	*A*8	82.2	84.8	83.4	73.8	77.0	75.3	85.6	85.3	85.4
*A*7	*O*1	77.1	82.5	79.7	67.6	73.2	69.9	79.4	81.9	80.3
*A*7	*O*2	78.5	81.6	80.0	70.6	74.2	71.8	80.6	81.4	80.9
*A*7	*O*3	81.0	84.9	82.9	71.3	75.7	73.3	83.1	83.8	83.0
*A*7	*A*1	71.0	74.4	70.9	62.1	64.8	60.3	75.8	78.0	75.7
*A*7	*A*2	70.5	78.2	73.8	62.0	69.1	64.3	75.3	79.6	76.5
*A*7	*A*7	72.6	77.4	74.0	62.2	67.0	63.1	74.5	77.5	75.3
*A*7	*A*8	81.9	84.7	83.3	73.7	77.2	75.3	82.8	82.7	82.4
*A*8	*O*1	77.0	82.4	79.6	67.2	72.7	69.6	80.8	84.4	81.7
*A*8	*O*2	78.4	81.5	79.8	70.4	74.0	71.7	82.0	84.7	83.0
*A*8	*O*3	80.9	84.9	82.8	71.0	75.2	72.9	84.7	87.6	85.6
*A*8	*A*1	71.0	74.2	70.7	61.4	64.3	60.0	73.7	75.0	71.5
*A*8	*A*2	70.4	78.1	73.7	61.7	68.8	64.1	75.0	80.1	75.9
*A*8	*A*7	72.6	77.2	73.7	62.2	66.6	62.5	75.7	78.2	74.9
*A*8	*A*8	81.9	84.9	83.4	73.6	77.0	75.1	84.2	86.5	85.2

TABLE 6.5: F1 score. Experiments on portability of models from one user to another. User-in vocabulary-out results are integrated in this table for convenience of analysis.

The results obtained for this part are presented in Table 6.5. The color visually duplicates the magnitude of F1 score specified in each cell. This is done for an easier comprehension of the table. We can make several observations on these results:

1. The used features show an impact on the results. Thus, standard features usually show better F1 than FastText word embeddings. One explanation is that standard features include 24 individual features covering different aspects of the linguistic and non-linguistic description of words, while the pre-trained FastText word embeddings rely only on the distribution of words and their similarity. Yet, the combination of two features (standard and FastText embeddings) usually improves overall results, sometimes going to up to 4.8 improvement of F-measure. We hypothesize that there exists a robust nonlinear dependency between some subsets of standard features and subword-level components of FastText word embeddings. Testing this hypothesis is the topic of our further research.

2. Recall values are always higher than Precision values. This means that the algorithm performs slightly better in returning most of the relevant results, than in providing correct class labels.

3. In each set of experiments, the best results are not obtained when the model of a given annotator is applied to own data. For instance, the *O1* model provides better results when tested on data from annotators *O2, O3* and *A8*. Similarly, the *A7* model shows better results when applied to data from annotators *O1, O2, O3* and *A8*. This is an important issue because it shows that the models acquired from one annotator can be successfully generalized over other annotators. The fact that the training and test model on the same annotator provides comparatively low results most probably signifies about overfitting, whereas this hypothesis should be properly tested.

4. Besides, it seems that the considered annotators form two clusters according to the classification of difficult medical words: one cluster with four annotators (*O1, O2, O3, A8*) and one cluster with three annotators (*A1, A2, A7*). We can observe a decrease up to -3 in F-measure for a combination of features (standard and FastText embeddings) compared to standard features only when cross-validating between users from different clusters (fig. 6.1). As we already explained in section 6.2.3, this issue may be related to the health literacy of annotators. This may indicate that the annotation models can be shared by people with similar skills and knowledge. Yet, to confirm this hypothesis, it is necessary to define the level of health literacy of annotators. This task is rather difficult because there are no existing tests created for computing the health literacy level for French-speaking healthy people. Another hypothesis is that some models may be more generalizable than other models. This hypothesis must also be verified with additional experiments.

6.4 FrnnMUTE impact study

With FrnnMUTE we experimented on using them both solely and in combination with standard features and FastText word embeddings as feature sets for classifying medical words using a decision tree. The detailed results of testing FrnnMUTE in user-in vocabulary-out and user-out vocabulary-in cross-validation scenarios are

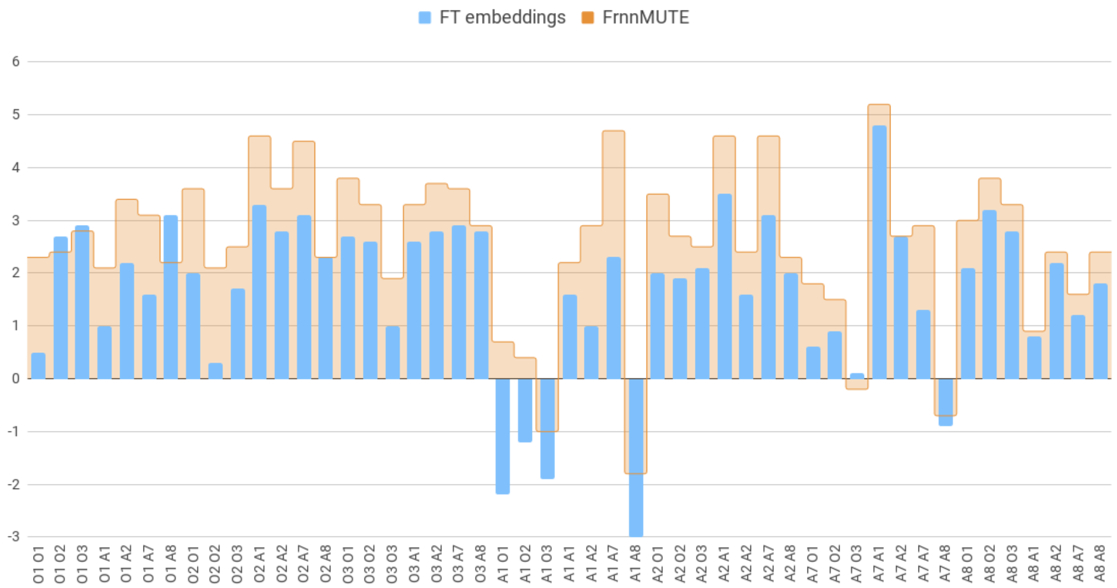


FIGURE 6.1: The difference of F1 score received for user pairs with a classification model built on combination of features (standard and based on deep learning) and on only standard features.

displayed in Table 6.8 (the logic of colors is the same as in Table 6.5 and is described in the previous section). To simplify the process of analyzing and comparing the results of this and the previous part, we aggregated the resulting F1 scores for combinations of a feature set and cross-validation scenario over all available users (Table 6.6). From the available results tables we can draw the following conclusions:

1. We observed that our FrnnMUTE solely perform better than FastText word embeddings solely in all cross-validation scenarios. FrnnMUTE provide the maximum among user pairs F1 score 79.5 versus 75.1 which provide FastText word embeddings in user-in vocabulary-out cross-validation; 82.4 versus 75.3 and 79.6 versus 74.9 for user-out vocabulary-in and user-out vocabulary-out scenarios correspondingly (Table 6.7).

2. FrnnMUTE's results have the smallest dispersion (3.8-3.9) among all considered "solo" feature sets types (4.8-5.3) when aggregating by all available users. This means that FrnnMUTE are more robust in generalizing information from user to user and between different subsets of vocabulary.

3. For user-in vocabulary-out and user-out vocabulary-out experiments the combination of standard features and FrnnMUTE in almost all cases show the best performance among all seven features sets. The improvement in F1 score over standard features with FastText embeddings can be observed on fig. 6.1. We can observe that the difference in F1 reaches 2.9 for some users pairs and the maximum improvement achieved by combining standard features with FrnnMUTE over using standard features only hits 5.2 in F-measure. This testifies that FrnnMUTE help standard linguistic and non-linguistic features to capture word understandability better than FastText embeddings.

4. The fact that the combination of all three types of feature sets performs insignificantly better or even worse than standard features with only FrnnMUTE can

be explained by overfitting of the classification model in the first case as the resulting feature vector has the biggest dimensionality.

μ +/- σ	user-in vocabulary-out	user-out vocabulary-in	user-out vocabulary-out
Standard features	77.7 +/- 5.2	77.7 +/- 4.9	77.6 +/- 4.9
FT emb	67.9 +/- 5.7	67.6 +/- 5.3	67.3 +/- 5.2
FrnnMUTE	75.1 +/- 3.9	77.1 +/- 3.9	74.5 +/- 3.9
Standard features + FT emb	78.9 +/- 5.1	79.5 +/- 4.6	77.1 +/- 4.6
Standard features + FrnnMUTE	80.0 +/- 5.1	80.3 +/- 4.3	78.6 +/- 4.4
Standard features + FT emb + FrnnMUTE	79.9 +/- 5.0	80.4 +/- 4.3	78.1 +/- 4.3

TABLE 6.6: Mean and standard deviation of F1 scores. Study of our FrnnMUTE's performance for word understandibility detection. The aggregation for μ and σ is performed through all user-pairs by cross-validation experiments and feature sets combinations. For words categorization with Only standard features/ Only FastText word embeddings/ Only FrnnMUTE a decision tree of depth 4 was trained. On all the rest of feature sets a decision tree of depth 9 was trained. The pair O1-O1 for user-in vocabulary-out cross validation is excluded from the aggregation for consistency of results for all users as FrnnMUTE was trained on annotations of O1.

Max F1 score	user-in vocabulary-out	user-out vocabulary-in	user-out vocabulary-out
Standard features	83.4	84.4	84.3
FT emb	75.1	75.3	74.9
FrnnMUTE	79.5	82.4	79.6
Standard features + FT emb	85.2	86.9	84.6
Standard features + FrnnMUTE	85.8	87.0	85.2
Standard features + FT emb + FrnnMUTE	85.8	87.4	85.2

TABLE 6.7: Maximum F1 score. The aggregation is performed through all user-pairs by cross-validation experiments and feature sets combinations. The pair O1-O1 for user-in vocabulary-out cross validation is excluded from the aggregation for consistency of results for all users as FrnnMUTE was trained on annotations of O1.

Train user	*Test user*	*Standard features*			*FrnnMUTE*			*Standard features + FrnnMUTE*		
		P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
*O*1	*O*1	77.2	82.5	79.7	76.1	81.0	78.4	79.3	84.9	82.0
*O*1	*O*2	78.6	81.7	80.1	78.8	80.7	79.6	82.2	82.9	82.5
*O*1	*O*3	81.2	85.0	83.0	80.7	82.6	81.3	85.3	86.5	85.8
*O*1	*A*1	71.0	74.7	71.2	71.4	74.5	71.5	75.0	75.7	73.3
*O*1	*A*2	70.6	78.4	74.0	72.0	77.3	73.6	76.5	80.2	77.4
*O*1	*A*7	72.6	77.5	74.2	74.4	78.1	75.2	77.5	79.4	77.3
*O*1	*A*8	82.3	84.9	83.5	81.2	82.2	81.5	85.5	85.8	85.7
*O*2	*O*1	77.0	82.2	79.1	77.4	82.3	79.5	82.0	85.4	82.7
*O*2	*O*2	78.9	82.0	80.0	76.1	79.3	77.6	80.8	83.9	82.1
*O*2	*O*3	81.1	85.4	83.0	79.6	83.1	81.2	84.6	87.5	85.5
*O*2	*A*1	71.1	72.1	68.2	71.3	73.7	70.1	74.8	76.2	72.8
*O*2	*A*2	70.8	77.3	72.7	72.9	77.4	73.1	76.4	80.5	76.3
*O*2	*A*7	72.7	75.6	71.8	73.9	77.1	73.7	76.9	79.6	76.3
*O*2	*A*8	83.0	86.2	84.4	81.5	83.8	82.4	85.8	88.1	86.7
*O*3	*O*1	77.4	82.8	79.7	78.9	82.5	80.0	82.5	85.8	83.5
*O*3	*O*2	79.0	82.2	80.2	79.0	81.3	79.7	82.9	84.7	83.5
*O*3	*O*3	81.2	85.5	83.2	77.0	80.6	78.7	85.4	87.3	85.1
*O*3	*A*1	71.8	73.3	69.5	72.6	72.8	69.4	75.5	75.9	72.8
*O*3	*A*2	71.2	78.0	73.5	74.0	77.1	73.1	77.2	80.8	77.2
*O*3	*A*7	73.2	76.5	72.9	74.5	76.3	73.1	77.6	79.4	76.5
*O*3	*A*8	82.6	85.8	84.1	81.4	83.6	82.3	86.2	88.0	87.0
*A*1	*O*1	77.2	82.5	79.8	78.4	80.6	78.7	80.2	82.4	80.5
*A*1	*O*2	78.6	81.6	80.1	78.3	79.1	78.3	80.6	81.2	80.5
*A*1	*O*3	81.2	84.9	82.9	80.2	80.2	79.3	82.8	82.9	81.9
*A*1	*A*1	70.9	74.7	71.3	70.0	73.5	70.4	71.5	76.8	73.5
*A*1	*A*2	70.5	78.3	74.0	73.4	77.4	73.8	76.6	80.4	76.9
*A*1	*A*7	72.6	77.5	74.2	74.9	78.7	76.0	78.1	81.5	78.9
*A*1	*A*8	82.2	84.8	83.5	80.3	79.7	79.3	82.7	82.1	81.7
*A*2	*O*1	77.3	82.6	79.8	79.6	81.7	80.5	82.4	84.5	83.3
*A*2	*O*2	78.6	81.6	80.1	79.4	79.9	79.6	82.7	83.0	82.8
*A*2	*O*3	81.2	84.9	83.0	81.3	81.7	81.2	85.2	85.9	85.5
*A*2	*A*1	70.9	74.6	71.2	73.9	75.4	73.3	77.4	77.7	75.8
*A*2	*A*2	70.6	78.4	74.0	72.1	75.7	71.6	76.4	80.3	76.4
*A*2	*A*7	72.6	77.5	74.2	75.6	78.1	76.1	79.1	80.6	78.8
*A*2	*A*8	82.2	84.8	83.4	81.8	81.5	81.4	85.8	85.6	85.7
*A*7	*O*1	77.1	82.5	79.7	79.1	81.1	79.2	80.9	83.7	81.5
*A*7	*O*2	78.5	81.6	80.0	78.7	79.3	78.6	81.1	82.4	81.5
*A*7	*O*3	81.0	84.9	82.9	80.5	80.5	79.6	82.9	84.0	82.7
*A*7	*A*1	71.0	74.4	70.9	72.5	76.3	73.4	75.1	79.1	76.1
*A*7	*A*2	70.5	78.2	73.8	73.1	77.4	73.8	75.6	80.4	76.5
*A*7	*A*7	72.6	77.4	74.0	70.4	76.1	73.1	74.3	80.1	76.9
*A*7	*A*8	81.9	84.7	83.3	80.5	80.0	79.6	83.0	83.3	82.6
*A*8	*O*1	77.0	82.4	79.6	78.2	82.4	79.6	81.8	85.3	82.6
*A*8	*O*2	78.4	81.5	79.8	79.3	81.9	80.2	82.9	85.2	83.6
*A*8	*O*3	80.9	84.9	82.8	80.9	83.7	81.7	85.0	88.1	86.1
*A*8	*A*1	71.0	74.2	70.7	72.0	73.1	69.5	74.5	75.2	71.6
*A*8	*A*2	70.4	78.1	73.7	73.1	77.3	72.9	75.5	80.4	76.1
*A*8	*A*7	72.6	77.2	73.7	73.5	76.5	73.0	76.2	78.7	75.3
*A*8	*A*8	81.9	84.9	83.4	78.0	81.2	79.5	84.3	87.5	85.8

TABLE 6.8: F1 score. Detailed study of FrnnMUTE's performance for word understandibility detection.

Chapter 7

Conclusions

7.1 Contribution

In this work, we considered the task of medical word understandability detection. This task was tackled as a multiclass classification problem, and we made the following contributions:

1. We broaden the methodology of working with the task by introducing two new types of cross-validation scenarios for model validation. Those scenarios are close to real-world situations:
    - when having the reference annotations from only a small group of users, we want our model to predict the understandability of the same set of words for all patients.
    - when the reference annotations are only available for a small group of users and a subset of all possible words, and we want our model to predict whether new users will understand new words.
2. For the first time, for the task of detecting French word understandability in the medical domain, we utilized FastText word embeddings as features. We found out that the embeddings solely as features are not enough for good word categorization as they do not capture the important linguistic and non-linguistic description of words (F1 score is between 69.3 and 72.3). However, adding FastText word embeddings to standard features results in a substantial improvement of classification model's performance when generalizing for unknown users: F1 score reaches 85.9 and the improvement in F1 score compared to results of classification using only standard features is up to 4.8 in absolute difference. We also found out that combining FastText word embeddings with standard features may provide a decrease in performance for user pairs with different levels of health literacy. Nonetheless, we consider the improvement of the model's generalization ability for most of the user pairs a positive issue as when scaling to the real-world situation it is important to be able to generalize annotations provided by a small set of users on the whole population.

    These results of applying FastText word embeddings for automatic word categorization on data from three annotators were published and presented on 1st International Workshop on Informatics & Data-Driven Medicine[1] (Pylieva et al., 2018).
3. Inspired by the encoder part of seq2seq models (Sutskever, Vinyals, and Le, 2014), we implemented a novel type of embeddings and called them Frnn-MUTE (French RNN Medical Understandability Text Embeddings). We found

[1] http://science.lpnu.ua/iddm-2018

out that compared with the case of using only standard features, the combination of our FrnnMUTE with standard features substantially improves the performance of classification model for all three generalization scenarios, both by unknown users and unknown words, providing up to 5.2 higher F1 score and reaching at maximum 87.0 F1 score for user-out vocabulary-in cross-validation (80.3 F1 score in average by user pairs for this cross-validation scenario). We also observed that the performance of standard features with FrnnMUTE is more robust and significantly better (up to 2.9 higher F-measure in user-out vocabulary-in cross-validation) than the performance of standard features with FastText word embeddings. This indicates that FrnnMUTE capture better the specifics of medical words required for identifying their understandability by different users than FastText word embeddings.

4. The combination of our FrnnMUTE with standard features slightly outperformed (by at most 1.3 F1 score in absolute difference) the results in paper (Grabar, Hamon, and Amiot, 2014), work from which we aimed to proceed in this work. Also, the maximum reached 87.0 F1 score in this work is comparable to results of the top-rank systems submitted on CWI 2018 Shared Task (discussed in 2.3.1) for monolingual (English) studies, although the formulations of tasks are not strictly the same.

   The FrnnMUTE trained as described in 5.1.3 is available for public access at GitHub[2] and can be used for scientific non-commercial purposes.

7.2 Future work

We have several directions for future work:

1. Currently, we use existing pre-trained word embeddings on Wikipedia and Web Crawl. We assume that training word embeddings on medical data may improve their impact on the categorization results.

2. After an analysis of results of the application of FastText word embeddings in the categorization task, we assumed the existence of a robust nonlinear dependency between some subsets of standard features and subword-level components of FastText word embeddings. We plan to test this hypothesis in further research.

3. While the annotations go forward, the annotators usually show *learning* progress in decoding the morphological structure of terms and their understanding (Grabar and Hamon, 2017). This progress is not taken into account in the current experiments, this is the topic of our future research.

[2] https://github.com/hpylieva/FrnnMUTE

Appendix A

Experiments with RNN as a direct classifier

Table A.1 represents the experiments we ran to choose a classification model for further extraction of FrnnMUTE from it. All experiments have the following in common:

- Computation engine: GPU Tesla K80.
- Class labels: annotator $O1$.
- Input data preprocessing: words are converted to lower case, Unicode converted to ASCII.
- Input size: 57 (the number of distinct ASCII characters in the training dataset).
- Number of hidden dimensions: 50.
- Output size: 3 (as we classify each word into tree classes as it is in annotations).
- Train samples choice: randomly; the number of samples is due to specified in the experiment.
- Loss: negative log-likelihood loss (NLLLoss).

Experiment Number	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Recurrent layer	LSTM	GRU	GRU	LSTM	LSTM	LSTM	LSTM	LSTM
Bidirectional	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Number of recurrent layers	1	2	2	2	2	2	2	2
Dropout	0.0	0.0	0.0	0.5	0.5	0.7	0.7	0.7
Test size	0.3	0.3	0.3	0.3	0.3	0.1	0.1	0.1
Time, min	20	41	41	62	122	33	42	42
Early stopping*	Yes	Yes	No	No	No	Yes	No	No
Number of epochs	4	11	20	10	16	7	10	12
Best score epoch	4	11	8	9	15	5	9	12
Accuracy on test	0.8078	0.8037	0.8059	0.8154	0.8100	0.8089	0.8121	0.8205
F1 Score	0.7806	0.7907	0.7872	0.7905	0.7856	0.7806	0.7894	0.7929

TABLE A.1: Experimenting with different configurations of RNN for words' classification.

To choose a model we tested the performance of a decision tree based classifier in user-in vocabulary-out cross-validation setting on FrnnMUTE the model provides.

Following this, we considered the model from experiment 7 the best, as it provided the highest average $F1$ score among seven words' annotations.

From Table A.2 we can see that nevertheless, the BiLSTM from experiment 8 has higher accuracy and F1 score than the LSTM from experiment 7, FrnnMUTE from the latter generalize better in the classification task solved with a decision tree. This effect is presumably due to overfitting of BiLSTM to the data it was trained on.

	LSTM from experiment 7				*BiLSTM from experiment 8*			
Annotator	*A (%)*	*P (%)*	*R (%)*	*F*	*A (%)*	*P (%)*	*R (%)*	*F*
O1	80.98	76.10	80.98	78.44	80.04	74.94	80.04	77.41
O2	79.26	76.06	79.26	77.56	79.06	75.91	79.06	77.40
O3	80.56	76.96	80.56	78.70	80.47	78.92	80.47	78.44
A1	73.49	70.04	73.49	70.36	71.88	67.46	71.88	68.81
A2	75.66	72.15	75.66	71.57	74.62	68.97	74.62	70.43
A7	76.12	70.35	76.12	73.07	74.88	69.39	74.88	71.69
A8	81.17	77.96	81.17	79.48	81.19	78.05	81.19	79.55
Average	78.18	74.23	78.18	75.60	77.45	73.38	77.45	74.82

TABLE A.2: Compare the performance of FrnnMUTE from LSTM and BiLSTM in words' classification task with a decision tree.

Bibliography

Aroyehun, Segun Taofeek et al. (2018). "Complex Word Identification: Convolutional Neural Network vs. Feature Engineering". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 322–327. DOI: 10.18653/v1/W18-0538. URL: http://aclweb.org/anthology/W18-0538.

Bingel, Joachim, Natalie Schluter, and Héctor Martínez Alonso (2016). "CoastalCPH at SemEval-2016 Task 11: The importance of designing your Neural Networks right". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1028–1033. DOI: 10.18653/v1/S16-1160. URL: http://aclweb.org/anthology/S16-1160.

Bojanowski, Piotr et al. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. URL: http://aclweb.org/anthology/Q17-1010.

Borst, A et al. (2008). "Lexically based distinction of readability levels of health documents". In: *MIE 2008*. Poster.

Brigo, F et al. (2015). "Clearly written, easily comprehended ? The readability of websites providing information on epilepsy". In: *Epilepsy & Behavior* 44, pp. 35–39.

Britz, Denny (2016). *Deep Learning for Chatbots, Part 1 – Introduction*. [Online; posted April 6, 2016]. URL: http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/.

Brooke, Julian, Alexandra Uitdenbogerd, and Timothy Baldwin (2016). "Melbourne at SemEval 2016 Task 11: Classifying Type-level Word Complexity using Random Forests with Corpus and Word List Features". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 975–981. DOI: 10.18653/v1/S16-1150. URL: http://aclweb.org/anthology/S16-1150.

Brownlee, Jason (2017). *What Are Word Embeddings for Text?* [Online; posted October 11, 2017]. URL: https://machinelearningmastery.com/what-are-word-embeddings/.

Che, Zhengping et al. (2016). "Recurrent Neural Networks for Multivariate Time Series with Missing Values". In: *CoRR* abs/1606.01865. arXiv: 1606.01865. URL: http://arxiv.org/abs/1606.01865.

Chen, Mia Xu et al. (2018). "The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation". In: *CoRR* abs/1804.09849. arXiv: 1804.09849. URL: http://arxiv.org/abs/1804.09849.

Chinchor, Nancy (1992). "MUC-4 Evaluation Metrics". In: *Proceedings of the 4th Conference on Message Understanding*. MUC4 '92. McLean, Virginia: Association for Computational Linguistics, pp. 22–29. ISBN: 1-55860-273-9. DOI: 10.3115/1072064.1072067. URL: https://doi.org/10.3115/1072064.1072067.

Chmielik, J and N Grabar (2011). “Détection de la spécialisation scientifique et technique des documents biomédicaux grâce aux informations morphologiques”. In: *TAL* 51.2, pp. 151–179.

Clercq, Orphée De et al. (2014). “Using the crowd for readability prediction”. In: *Natural Language Engineering* 20, pp. 293–325.

Collins-Thompson, Kevyn (2014). “Computational assessment of text readability: A survey of current and future research”. In: *International Journal of Applied Linguistics* 165.2, pp. 97–135. DOI: https://doi.org/10.1075/itl.165.2.01col.

Collobert, Ronan et al. (2011). “Natural Language Processing (almost) from Scratch”. In: *CoRR* abs/1103.0398. arXiv: 1103.0398. URL: http://arxiv.org/abs/1103.0398.

Côté, Roger A. et al. (1993). *The Systematised Nomenclature of Human and Veterinary Medicine: SNOMED International*. Northfield: College of American Pathologists.

De Hertog, Dirk and Anaïs Tack (2018). “Deep Learning Architecture for Complex Word Identification”. In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 328–334. DOI: 10.18653/v1/W18-0539. URL: http://aclweb.org/anthology/W18-0539.

Eysenbach, Gunther (2007). “Poverty, Human Development, and the Role of eHealth”. In: *J Med Internet Res* 9.4, pp. 34–4.

Flesch, R (1948). “A new readability yardstick”. In: *Journ Appl Psychol* 23, pp. 221–233.

Gala, N, T François, and C Fairon (2013). “Towards a French lexicon with difficulty measures: NLP helping to bridge the gap between traditional dictionaries and specialized lexicons”. In: *eLEX-2013*.

Goeuriot, L, N Grabar, and B Daille (2008). “Characterization of scientific and popular science discourse in French, Japanese and Russian”. In: *LREC*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Grabar, N, S Krivine, and MC Jaulent (2007). “Classification of Health Webpages as Expert and Non Expert with a Reduced Set of Cross-language Features”. In: *Ann Symp Am Med Inform Assoc (AMIA)*, pp. 284–288.

Grabar, Natalia, Emmanuel Farce, and Laurent Sparrow (2018). “Study of readability of health documents with eye-tracking and machine learning approaches”. In: *Int Conf on Healthcare Informatics (ICHI)*. Poster, pp. 1–2.

Grabar, Natalia and Thierry Hamon (2016). “A large rated lexicon with French medical words”. In: *LREC (Language Resources and Evaluation Conference)*, pp. 1–12.

— (2017). “Understanding of unknown medical words”. In: *Proceedings of the Biomedical NLP Workshop associated with RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., pp. 32–41. DOI: 10.26615/978-954-452-044-1_005. URL: https://doi.org/10.26615/978-954-452-044-1_005.

Grabar, Natalia, Thierry Hamon, and Dany Amiot (2014). “Automatic diagnosis of understanding of medical words”. In: *EACL PITR Workshop*, pp. 11–20.

Gunning, Robert (1973). *The technique of clear writing*. New York, NY: McGraw Hill.

Harris, Zellig (1954). “Distributional Structure”. In: <i>WORD</i> 10.2-3, pp. 146–162. DOI: 10.1080/00437956.1954.11659520. URL: https://doi.org/10.1080/00437956.1954.11659520.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hyafil, L. and R. L. Rivest (1976). “Constructing optimal binary decision trees is NP-complete”. In: *Information Processing Letters* 5.1, pp. 15–17. URL: https://www.sciencedirect.com/science/article/pii/0020019076900958.

Jiang, Fei et al. (2017). “Artificial intelligence in healthcare: past, present and future”. In: *Stroke and Vascular Neurology* 2.e000101. DOI: :10.1136/svn-2017-000101. URL: https://svn.bmj.com/content/svnbmj/2/4/230.full.pdf.

Jucks, R and R Bromme (2007). “Choice of words in doctor-patient communication: an analysis of health-related internet sites”. In: *Health Commun* 21.3, pp. 267–77.

Kauchak, David (2013). “Improving Text Simplification Language Modeling Using Unsimplified Text Data”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1537–1546. URL: http://aclweb.org/anthology/P13-1151.

Kincaid, JP et al. (1975). *Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel*. Tech. rep. Memphis, TN: Naval Technical Training, U. S. Naval Air Station.

Kohavi, Ron (1995). “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: Morgan Kaufmann, pp. 1137–1143.

— (1998). “Glossary of terms”. In: *Special Issue on Applications of Machine Learning and the Knowledge Discovery Process* 30.271, pp. 127–132. URL: https://ci.nii.ac.jp/naid/10018512237/en/.

Kokkinakis, D and M Toporowska Gronostaj (2006). “Comparing Lay and Professional Language in Cardiovascular Disorders Corpora”. In: *WSEAS Transactions on BIOLOGY and BIOMEDICINE*. Ed. by Australia Pham T. James Cook University, pp. 429–437.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Kuhn, Max and Kjell Johnson (2014). *Applied Predictive Modeling*. Springer Publishing Company, Incorporated. ISBN: 978-1-4614-6848-6.

Li, Fei-Fei, Andrej Karpathy, and Justin Johnson (2016). *CS231n: Convolutional Neural Networks for Visual Recognition*. URL: http://cs231n.stanford.edu/.

Malmasi, Shervin, Mark Dras, and Marcos Zampieri (2016). “LTG at SemEval-2016 Task 11: Complex Word Identification with Classifier Ensembles”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 996–1000. DOI: 10.18653/v1/S16-1154. URL: http://aclweb.org/anthology/S16-1154.

McCray, Alexa T. (2005). “Promoting Health Literacy”. In: *Journal of the American Medical Informatics Association* 12.2, pp. 152–163. DOI: 10.1197/jamia.M1687. eprint: /oup/backfile/content_public/journal/jamia/12/2/10.1197/jamia.m1687/2/12-2-152.pdf. URL: http://dx.doi.org/10.1197/jamia.M1687.

Mikolov, T et al. (2013a). “Distributed Representations of Words and Phrases and their Compositionality”. In: *NIPS*.

Mikolov, T et al. (2013b). “Efficient Estimation of Word Representations in Vector Space”. In: *Workshop at ICLR*.

Mikolov, Tomas et al. (2017). “Advances in Pre-Training Distributed Word Representations”. In: *CoRR* abs/1712.09405. arXiv: 1712.09405. URL: http://arxiv.org/abs/1712.09405.

Miller, T et al. (2007). "A Classifier to Evaluate Language Specificity of Medical Documents". In: *HICSS*, pp. 134–140.

Namer, F (2000). "FLEMM : un analyseur flexionnel du français à base de règles". In: *Traitement automatique des langues (TAL)* 41.2, pp. 523–547.

Namer, Fiammetta and Pierre Zweigenbaum (2004). "Acquiring meaning for French medical terminology: contribution of morphosemantics". In: *Ann Symp Am Med Inform Assoc (AMIA)*. San-Francisco.

Ng, Andrew (2012). *CS229 Lecture notes - Supervised learning*. URL: http://cs229.stanford.edu/notes/cs229-notes1.pdf.

Olah, Christopher (2015). *Understanding LSTM Networks*. [Online; posted August 27, 2015]. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Oregon Practice Center (2008). *Barriers and Drivers of Health Information Technology Use for the Elderly, Chronically Ill, and Underserved*. Tech. rep. Agency for healthcare research and quality. Oregon Evidence-based Practice Center.

Paetzold, Gustavo and Lucia Specia (2016a). "SemEval 2016 Task 11: Complex Word Identification". In: *SemEval@NAACL-HLT*. The Association for Computer Linguistics, pp. 560–569.

— (2016b). "SV000gg at SemEval-2016 Task 11: Heavy Gauge Complex Word Identification with System Voting". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 969–974. DOI: 10.18653/v1/S16-1149. URL: http://aclweb.org/anthology/S16-1149.

Patel, V, T Branch, and J Arocha (2002). "Errors in interpreting quantities as procedures : The case of pharmaceutical labels". In: *Int Journ Med Inform* 65.3, pp. 193–211.

Pearson, Jennifer (1998). *Terms in Context*. Vol. 1. Studies in Corpus Linguistics. Amsterdam/Philadelphia: John Benjamins.

Plank, Barbara, Anders Søgaard, and Yoav Goldberg (2016). "Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss". In: *CoRR* abs/1604.05529. arXiv: 1604.05529. URL: http://arxiv.org/abs/1604.05529.

Podgorelec, Vili et al. (2002). "Decision Trees: An Overview and Their Use in Medicine". In: *Journal of medical systems* 26, pp. 445–63. DOI: 10.1023/A:1016409317640.

Poprat, M, K Markó, and U Hahn (2006). "A Language Classifier that Automatically Divides Medical Documents for Experts and Health Care Consumers". In: *Int Congress of the European Federation for Medical Informatics*. Maastricht, pp. 503–508.

Pylieva, Hanna et al. (2018). "Improving Automatic Categorization of Technical vs. Laymen Medical Words using FastText Word Embeddings". In: pp. 93–102. URL: http://ceur-ws.org/Vol-2255/paper9.pdf.

Quinlan, J. R. (1986). "Induction of Decision Trees". In: *MACH. LEARN* 1, pp. 81–106.

Quinlan, JR (1993). *C4.5 Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Ronzano, Francesco et al. (2016). "TALN at SemEval-2016 Task 11: Modelling Complex Words by Contextual, Lexical and Semantic Features". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1011–1016. DOI: 10.18653/v1/S16-1157. URL: http://aclweb.org/anthology/S16-1157.

Sackett, David L et al. (1996). "Evidence based medicine: what it is and what it isn't". In: *BMJ* 312.7023, pp. 71–72. ISSN: 0959-8138. DOI: 10.1136/bmj.312.7023.71.

eprint: https://www.bmj.com/content. URL: https://www.bmj.com/content/312/7023/71.

Schmid, H (1994). "Probabilistic Part-of-Speech Tagging Using Decision Trees". In: *Int Conf on New Methods in Language Processing*, pp. 44–49.

Sebastiani, Fabrizio (2002). "Machine learning in automated text categorization". In: *ACM Computing Surveys* 34.1, pp. 1–47. ISSN: 0360-0300. DOI: http://doi.acm.org/10.1145/505282.505283.

Sherman, Lucius (1893). *Analytics of literature: A manual for the objective study of English prose and poetry*. Boston: Ginn and Co.

Si, Luo and James P. Callan (2001). "A Statistical Model for Scientific Readability." In: pp. 574–576. DOI: 10.1145/502585.502695.

Stehnii, Anatolii (2017). "Generation of code from text description with syntactic parsing and Tree2Tree model". MA thesis. Lviv: Ukrainian Catholic University.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, pp. 3104–3112. URL: http://dl.acm.org/citation.cfm?id=2969033.2969173.

Tran, TM et al. (2009). "Internet et soins : un tiers invisible dans la relation médecin/-patient ?" In: *Ethica Clinica* 53, pp. 34–43.

Vander Stichele, RH (1999). "Promises for a measurement breakthrough". In: *Drug regimen compliance. Issues in clinical trials and patient management*. Ed. by John Wiley & Sons. JM Metry and UA Meyer, pp. 71–83.

Weng, Lilian (2017). *Learning Word Embedding*. [Online; posted October 15, 2017]. URL: https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html.

Williams, MV et al. (1995). "Inadequate functional health literacy among patients at two public hospitals". In: *JAMA* 274.21, pp. 1677–1682.

Yaneva, V, I Temnikova, and R Mitkov (2015). "Accessible texts for autism: An eye-tracking study". In: *Int ACM SIGACCESS Conference on Computers & Accessibility*. Ed. by ACM, pp. 49–57.

Yimam, Seid Muhie et al. (2018). "A Report on the Complex Word Identification Shared Task 2018". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 66–78. DOI: 10.18653/v1/W18-0507. URL: http://aclweb.org/anthology/W18-0507.

Zeng-Treiler, Q et al. (2007). "Text characteristics of clinical reports and their implications for the readability of personal health records". In: *MEDINFO*. Brisbane, Australia, pp. 1117–1121.

Zhang et al. (2015). "Character-level Convolutional Networks for Text Classification". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, pp. 649–657. URL: http://dl.acm.org/citation.cfm?id=2969239.2969312.

Zheng, W, E Milios, and C Watters (2002). "Filtering for medical news items using a machine learning approach". In: *Ann Symp Am Med Inform Assoc (AMIA)*, pp. 949–53.