Crowd Sourcing as an Improvement of N-Grams Text Document Classification Algorithm

1st Petr Šaloun
Palacky University Olomouc
Krizkovskeho 511/8
CZ-771 47 Olomouc, Czech republic
Email: petr.saloun@upol.cz

2nd David Andrešič
Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava
Ostrava, Czech republic
Email: david.andresic.st@vsb.cz

3rd Barbora Cigánková
Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava
Ostrava, Czech republic
Email: barbora.cigankova.st@vsb.cz

4th Ioannis Anagnostopoulos
Computer Science and Biomedical Informatics Dpt.
University of Thessaly
Lamia, Greece
Email: janag@dib.uth.gr

5th Lenka Krhutová
Faculty of Social Studies
University of Ostrava
Ostrava, Czech republic
Email: lenka.krhutova@osu.cz

***Abstract*—A common task in a world of natural language processing is text classification useful for e.g. spam filters, documents sorting, science articles classification or plagiarism detection. This can still be done best and most accurately by human, on the other hand, we can often accept certain error in the classification in exchange for its speed. Here, natural language processing mechanism transforms the text in natural language to a form understandable by a classifier such as K-Nearest Neighbour, Decision Trees, Artificial Neural Network or Support Vector Machines. We can also use this human element to help automated classification to improve its accuracy by means of crowdsourcing.**

This work deals with classification of text documents and its improvement through crowdsourcing. Its goal is to design and implement text documents classifier prototype based on documents similarity and to design evaluation and crowdsourcing-based classification improvement mechanism. For classification the N-grams algorithm has been chosen, which was implemented in Java. Interface for crowdsourcing was created using CMS WordPress. In addition to data collection, the purpose of interface is to evaluate classification accuracy, which leads to extension of classifier test data set, thus the classification is more successful.

We have tested our approach on two data sets with promising preliminary results even across different languages. This led to a real-world implementation started at the beginning of 2019 in cooperation of two universities: VŠB-TUO and OSU.

***Index Terms*—Classification, text documents, natural language processing, documents similarity, N-grams, crowdsourcing, WordPress, Java, PHP**

I. Introduction

Computers have became a common part of our lives. From this reasons, user experience is becoming a pressing issue. One thing that helps a lot in this manner is natural language processing (NLP), which is usually used in for example information extraction tasks or text classification, where it helps to automate and speed up the classification process.

Despite the progress in text classification, humans are still more accurate, which opens a space for human-assisted classification, e.g. by means of having human as a reference system. Here, it is also possible to use the collective intelligence of multiple people for such task, which is called crowdsourcing that is experiencing boom in the last years.

In this work, we describe our experience and preliminary results of Czech, Slovak and English text documents classification. We also offer a general overview of current algorithms for text classification, lemmatization and stemming focused on czech language. We also summarize crowdsourcing advantages, methodology and comparison with other approaches. We tested the approach on two data sets and compared with crowdsourcing approach. For this, we have developed an application where the communication with the crowd is done via a web Content Management System (CMS) built on top of WordPress. The result application is supposed to help caretakers for people with stroke which is our primary case study.

II. Text Document Classification Overview

The main goal of text classification is to assign the given text to some of the pre-defined classes. In the area of text mining, it is also a process of automatic learning of categorization schemas used for direct classification of new, uncategorized

documents [1]. Some approaches use different forms of document similarity metric, such as cosine similarity. This metric is then used in learning as well in classification phases. Before the classification itself, it is necessary to perform two steps:

- Transformation of the document to a form that can be parsed. This includes removal of stop words, tags and other pre-processing (see section III).
- Extraction of text properties that are then evaluated and their weight is calculated. These properties are then represented as vectors describing a presence of words or syntactic unit [1].

Many classifiers use a bag-of-words (BOW) approach for text representation [1]. It is a simplified text representation used mostly for NLP and information extraction where the document is transformed to a set of individual words without grammar structures and words order, but still containing possible words duplicity. During the classification, an occurrence frequency for each word in the bag is calculated so it can be then used as an input for classifier during training.

Today classifiers use either statistical approaches or machine learning and can be divided into two categories: supervised and unsupervised. Further text in this section describes today most common algorithms such as decision trees, N-grams, artificial neural networks and Bayes classifier [1].

A. Naive Bayes Classifier

Naive Bayes Classifier is a probability-based classifier built on top of Bayes theorem (described for example in [2]) saying how conditional probability of some event relates to an opposite conditional probability. Bayes classifier assumes that presence or absence of some attribute of the given class is not dependent on presence or absence of some other attribute [1]. In other words it expects that attributes are not dependent on each other. The advantage of Bayes classifier is that it performs well with smaller training data set to determine statistical parameters.

B. TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is often used for term-weighting (evaluation of individual text attributes). It is a statistical metric that measure an importance of words in the given document [3]. Term Frequency stands for count of the given word in the document divided by the total count of all words in the document. This normalization is done to eliminate the advantage of long documents in such calculation. The Inverse Document Frequency represents the importance of individual words. It is characterized as a logarithm of count of all documents divided by count of documents containing the given word [3]:

- TF(t) = (count of t in the document) / (total count of words in the document)
- IDF(t) = (total count of documents / count of documents that contain the t)

Matching documents will then have a high frequency of the given word that is also not so much present in other documents. One of the major disadvantages of TF-IDF is its ignorance of key semantic connections between words because it compares documents only based on frequency of individual words. Still, different variations of TF-IDF are often used in search engines for document ranking [1].

C. Latent Semantic Analysis (LSA)

LSA (also known as Latent Semantic Indexing - LSI) is a technique used for NLP. It is based on analysis of relationship between set of documents and words contained in them. In contrast to classic natural language processing or artificial intelligence approaches, LSA is not using any human-created dictionary, knowledge base, grammar or syntactic parser. The input of LSA is just a text divided into meaningful parts such as sentences or paragraphs [4]. LSA uses mathematical approach called Singular Value Decomposition (SVD). It is a method of linear algebra in which a regular matrix is decomposed to 3 smaller matrices such that matrix multiplication of these matrices must return the original matrix. The whole process is described for example in [5].

D. Support Vector Machines

Support Vector Machine (SVM) is a supervised machine learning method that is usually used in binary classification and regression analysis. In is based on a concept of decision planes that defines decision borders [6]. SVM uses a mechanism called hyperplanes in multidimensional space which divides objects of individual classes. The main idea of SVM is to allow linear division of objects of different classes using object transformation that is being done by mathematical functions called kernel function [6]. It is then crucial to find the most fitting hyperplane (plane with maximal margin), that is, find the place in which the the distance between closest points to the plane is as large as possible. In order to describe the hyperplane, we need just points that lies at the edge of maximal margin. These points are called support vectors [7]. Other points are not relevant to the hyperplane. SVM method is therefore capable to find those training samples which are most relevant to finding the hyperplane. The size of the training set required for classifier learning is therefore much smaller [7]. We recognize several types of SVM that differ by used iterative algorithm for error function minimization. They are described for example in [6].

E. N-grams

N-gram is defined as a tuple of N items that belongs to some sequence of e.g. words or characters. Sequence of two items is called bigram, sequence of three items then trigram. From four, it is called generally as N-gram. N-grams are usually used for text representation where words are used in the sequence. Another possible usage is document classification based on document similarity. During the classification, sequence of e.g. characters is used. The beginning and the end of the word is then marked by some special character such as underscore [8].

In general, a set of N-grams for a string of length k will contain $k+1$ N-grams. Great advantage of classification using

N-grams is its independence on document language, because there is no need for text pre-processing such as stemming or lemmatization. Another advantage is a certain tolerance to grammar errors and typos.

On the other hand, a large number of generated N-grams can be considered as a drawback. On the other hand, this can be reduced by e.g. removing stop words or by using stemming or lemmatization (or some other text length reduction), but by doing this, we lose the advantage of language independence.

In [20] authors for example used character N-grams and unigram indices for Twitter tweets classification. They confirmed language independence but also conclude that although character n-grams of 4-6 characters length leads to classification models with decent performance, the manually indicated tokens (a.k.a. crowdtagging) combined with a Decision Tree classifier outperform any other feature set-classification algorithm combination [20].

III. Processing of Texts Written in Natural Language

NLP software requires consistent knowledge base such as large dictionary, grammar rules, ontology and synonyms etc. [10]. NLP process consists of several phases using different methods to "decrypt" multiple language unclarities, e.g. tagging of part of the speech or understanding and recognition of the natural language [10]. These phases can be [9]: *morphological analysis*, *syntax analysis* and *semantic analysis*. Morphological analysis processes a single word as the smallest atomic unit. Using dictionary, it assigns a basic form to a word, word class and other morphological categories. Syntax analysis, on the other hand, processes whole sentences and formal description of their structures. Semantic analysis determines the meaning of word or a broader sentence. From these methods, morphological analysis is the one most explored and most algorithmizable [9]. On the other hand, semantic analysis is generally most difficult due to words homonymy.

Before almost every text processing, several pre-processing steps must be done, such as transformation to a lower-case form, removing of special characters, stop words and tokenization. Another usual steps are stemming and lemmatization.

A. Stemming and Lemmatization

Stemmers and Lemmatizers are attempting to find the common base or root of each word in the text. These tools are useful for e.g. counting the frequency of words in text because they allow to unite different forms on a words with the same meaning. Stemmers are working with individual words without context and thus cannot distinguish between different meaning of words. They are simply cutting prefixes and suffixes (and leaving just stems). For more details, please see e.g. [11]. On the other hand, lemmatization is working with morphological analysis of words. Lemmatization tools are working with grammar rules for the document language. More details can be found in e.g. [12].

1) Stemmers and Lemmers for Czech Language: Czech language is in general one of those more difficult for stemming and lemmatization. Czech language uses a lot of prefixes and has more complex inflection. Due to this there are not many usable frameworks or software libraries.

One solution offers Apache Lucene[1]. This search engine offers Czech language analyzer that contains set of Czech stop words, stemmer and tokenizer that can be enhanced by filters for e.g. lower-case transformation. The only disadvantage is the absence of Czech lemmatizer. This can be compensated by Czech morphological analyzer developed by Masaryk university[2] called Majka[3]. In its base settings, it assigns to each word [13]: basic form and grammar mark, all words related to the same lemma and all possible words with diacritic.

IV. Crowdsourcing

The concept of crowdsourcing can be defined as a business practice where the given activity is outsourced to a crowd [14]. Another definition can be found in [15] where author says that crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. The most important part of this definition is the undefined network of people. Everyone can then get task assigned to him or her. The only selection that is done in such process is selection of achieved results. Results are also often just aggregated.

Crowdsourcing theoretical roots were defined in [22]. It is based on an idea of collective intelligence. This concept can be understood as "all together we are smarter than just one of us" [16]. It is a concept also known as wisdom of the crowd. In [14] authors attempts to answer the 8 basic questions about crowdsourcing. As for advantages of crowdsourcing, we can name for example releasing core company employees for other work and lower expenses. A nice description of crowdsourcing pros can be found in [17]. One of the most difficult tasks in crowdsourcing usage is finding the right crowd motivation [18].

A. Examples of Use

Several large companies successfully used the crowdsourcing in real world applications [23]:

- **Waze** - Application for collecting traffic situation data.
- **Lego** - People suggests a new product and vote.
- **Samsung** - To find innovative solutions for their products.
- **Lays** - Same as **McDonald's** to create a new taste.
- **Greenpeace** - To get best sarcastic phrases for its campaign against Shell.

In [20] authors used crowdsourcing to obtain tokens for sentiment analysis of tweets and used them as a feature set which turned out to perform best in compare to other feature sets established by other means (e.g. N-grams). Similarly

[1]https://lucene.apache.org/

[2]https://www.muni.cz/

[3]https://www.muni.cz/vyzkum/publikace/935762

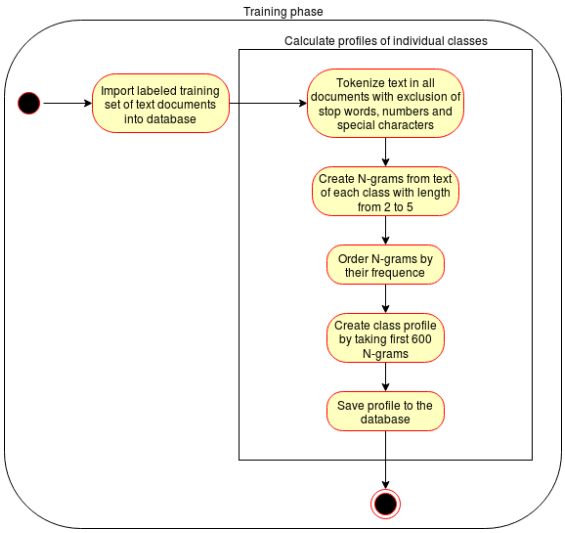

Fig. 1. Classifier training phase.

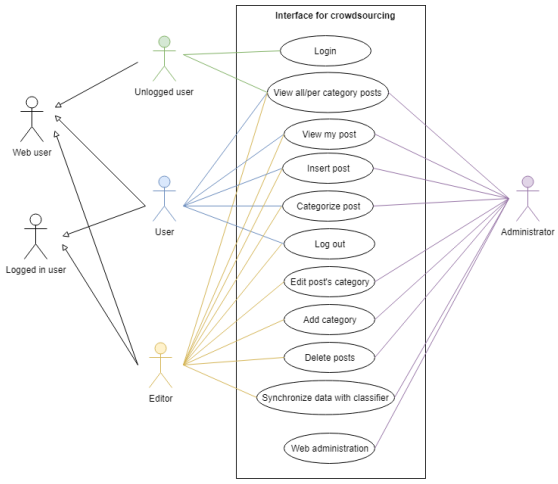

Fig. 2. Prototype use case diagram.

in [21] authors compared various kinds of low-level features, including those extracted through deep learning and conclude that keywords suggested by the crowd (called crowd lexicon-herein that are based on crowdtagging), established through a crowdsourcing platform can be effectively used for training sentiment classification models for short texts (tweets and Facebook comments) and that those models are at least as effective as the ones that are developed through deep learning or even better [21].

V. Prototype Description

From a large number of available algorithms we eventually chose the classification using N-grams – mostly for the easiness of its implementation but also for language independence. The implementation is based on N-gram-based text categorization described in [8] and consists from two phases: training (see Fig. 1) and classification. It the classification phase, the profile of unknown document is calculated (similar to the training phase, just for a single document). Then the distance between unknown document profiles and profiles in database are calculated using out-of-place method. And at last, unknown document is assigned with a class with shortest distance.

In opposite to [8] our classifier utilizes a reduction of count of words in document by removing the stop words. Using this reduction, it is not necessary to start in the profile class at the position 300 (as suggested in [8]) but it is possible to start from the beginning of the list. Also, our classifier works with longer profiles, mostly because of planned classification of psychological text. Their classes have usually a very thin border so we can expect the need of more N-grams.

Beside the classification, our application also determines key words of each class. These key words will then be displayed to selected users with a kind request to use them in their contribution. By showing key words only to some users, we create two user groups that will serve as referential groups to confirm the following hypothesis:

- Classification will perform better if contributions for classification contain pre-defined key words.

The calculation of key words is realized by TF-IDF algorithm modified (in respect to [19]) to class purpose. The calculation will look like following:

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{1}$$

where n_{ij} is frequency of term i in documents of class j.

$$IDF_i = log(\frac{|D|}{|d : t_i \in d|}) \tag{2}$$

where t_i is term and D is set of all classes. TF_{ij} is quotient of term frequency n_{ij} to count of all word in documents of the given class. IDF_i is then logarithm of quotient of classes count to count of classes containing term t_i.

Five key words with greatest weight per category from those obtained by this method are selected and stored to database. As it can be seen at Fig. 2, we distinguish several user roles for **crowdsourcing** user interface. Logged users can insert their contributions (some of them will be kindly asked to use pre-selected keywords based on selected contribution category). These contributions will be then classified by our classifier and in case of discrepancy, user will correct the category. This contribution will then be added to the training set (by Editor via manual data synchronization) in order to improve classifier's performance. For entering contributions we also plan to prepare a sort of tag cloud containing known synonyms of already entered words (based on already-trained model). This tag cloud will be updated on each key press and synonyms suggestions it should lead the user to enter a contribution that still has the same meaning but is classified more accurately and confidently (see Fig. 3). For this we already posses term frequencies for each class and establishing a dictionary of synonyms (including both laical and expert words), but current pandemic situation slowed down our progress as we cannot collect so many documents for training as we expected.

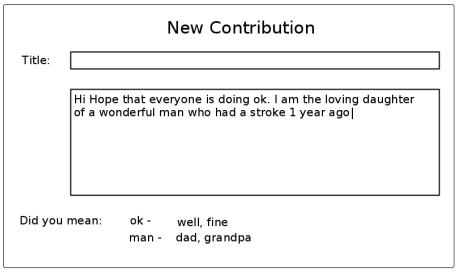

Fig. 3. Wireframe of suggestions made by tag cloud.

Class	*Accuracy*
English	20/20
Slovak	20/20
Czech	20/20
Sum	60/60

TABLE I
CLASSIFICATION ACCURACY FOR LANGUAGE DATA SET.

VI. PRELIMINARY RESULTS

Our classifier was tested on two data sets and then briefly with use of crowdsourcing.

A. Data Sets

Both data sets contained X text documents in Y classes. Each was then split into training and test set. After processing of each set, the classification accuracy was evaluated.

1) Language Data Set: First data set contains texts in different languages. The aim of this data set was to confirm language independence of the classifier. We used Czech, Slovak and English texts here. Each category contained 40 texts with 60 up to 20 words. Both training and test data set contained 20 texts for each class. The source of the data is DATAKON conference (2010, 2012, 2013 and 2014). Texts are therefore technical. Selection on texts was random. As we can see in results (Table I), our classifier successfully classified all 60 documents in test data set. Perfect accuracy was expected for English, but we expected worse numbers for Czech and Slovak that are very similar to each other.

The language independence of N-grams method was therefore confirmed. Further work could include adding more languages such as Polish or Russian language that are similar to Czech and Slovak as well. Also, since our data set was quite small, we could add some more data.

2) Psychological Data Set: Contrary to the first data set, this data set contains not so balanced count of texts for each class. Its aim was to investigate how the classifier will perform with not so well structured data. Also, these texts contains psychological topic. They are sorted to classes which borders are not so clear as in case of previous data set. These texts are often difficult to classify even by human. The expected accuracy of classification therefore was not high. The data set contained 87 documents in following classes:

- personal issues, illness etc. (63 documents),
- work, finances, school (8 documents),
- romantic, family and employment relationships (14 documents).

Class	*Accuracy*
Personal issues, illness	12/13
Work, finances, school	1/2
Romantic, family and employment relationships	2/3
Sum	15/18

TABLE II
CLASSIFICATION ACCURACY FOR PSYCHOLOGICAL DATA SET.

Training set contained 80% of randomly chosen documents, 20% were left as a test set. As it can be seen in Table II, in each class, one document was classified incorrectly. One personal issue was marked as an relationship issue. From calculated profile distances we can assume that this error could be caused by text length that corresponds to the topic. The entire text contained 150 words, but the introduction is about a different topic, which confuses the classifier that may not have enough of N-grams to compare with other classes profiles. Two more documents were classified as personal issues, illness etc. Their content from work, finances and school class is close to a border of two topics (school and personal issues) that can be also seen in a small distance between document and class profile. This document can also be marked as difficult to be classified by human. The last document is again a border one. Here, unexpectedly, the difference in profile distances is quite high and correct class is the most distant one. A possible improvement could be a better structure of data set.

3) Crowdsourcing: The classifier accuracy was also tested by implemented crowdsourcing interface. Our crowd contained people from OSU[4] and VŠB-TUO[5] universities.

Topics of contributions inserted into the interface were suggested as life of non-formal care takers and its influencing as a consequence of care taking. Based on this, following classes were created: *motivation for care taking*, *benefits and consequences of care taking*, *support of care takers* and *needs of care takers*. Texts of first class contains subtopics such as what motivates a care taker or what does not allow him to let the person to a institutional care. Second class is about texts containing consequences for care takers. Third class is about what could help or helps care takers in doing their job. Last class is about what care takers miss in their lives. Training set was provided by OSU. It consisted 180 one- or two-sentence texts classified into 4 categories. Crowd that create texts for classification (and also performing classification accuracy testing) consisted from students of Faculty of Medicine of OSU. During a test phase, correct class was assigned to the text in case of error. Another aim of this work was also suggest an approach that will lead to increased classification quality using crowdsourcing. The suggested approach was extension of training data set for the classifier. The data set was extended mostly by incorrectly classified contributions. Until now, 8 contributions were added (2 into each class) by

[4]https://www.osu.cz/
[5]https://www.vsb.cz/

a single author, which means a similar dictionary. Texts were added in 2 phases. First, one contribution was added to each class that led to zero-percent accuracy caused probably by a difference in contribution text nature and training data. After data synchronization, second phase was done in the same way with accuracy of 50%. We can see an increased accuracy and learning of classifier, yet we cannot make any conclusions due to small amount of data and the single author. During this work, a hypothesis about increasing classification quality using defined key words was mentioned. To prove this, users of crowdsourcing interface were divided into 2 groups: with key words and without them. Nevertheless, in our experiment all contributions were added by a single author so we cannot make nay conclusions yet.

VII. Conclusion

The aim of this work was to create a prototype of text document classifier based on text document similarity with further usage of crowdsourcing in order to increase classification accuracy. After an analysis of classification algorithms, N-grams algorithm was chosen, mainly for its language independence but also for is easy implementation. The classifier was then connected with the crowdsourcing interface. The accuracy was tested on two data sets and then by crowdsourcing interface. With first data set classifier performed excellently, which proves its language independence. Second data set with psychological data also performed very good, because incorrect classifications would be difficult even for human. Eventually, classifier accuracy was left to the users themselves using our crowdsourcing interface. In order to improve the accuracy, extending the training data set (especially with incorrectly classified texts) was suggested. With respect to a small number of contributions it is not possible to make further conclusions about classifier accuracy. Nevertheless even with such small sample we can see a growing trend. These results, on the other hand, shall not be generalized because all contributions were made by a single author. In this work, a hypothesis about increasing classification accuracy using key words from available data was mentioned. Nevertheless, in respect to a small number of available contributions, we cannot make any conclusions yet. Further work will include proving the hypothesis and data synchronization of the system (now performed every 3 hours). Despite having just small data set, our proof of concept and preliminary results has led to the real-world implementation that is now being done in cooperation of VŠB-TUO and OSU universities. We plan to prove the hypothesis and incorporate crowdsourcing in a real application.

Acknowledgment

The following grants are acknowledged for the financial support provided for this research: TACR No. TL01000299 and TACR No. TL02000050.

References

[1] KARMAN, S. Senthamarai; RAMARAJ, N. Similarity-Based Techniques for Text Document Classification. Int. J. SoftComput, 2008, 3.1: 58-62.

[2] OPITKA, P.; ŠMAJSTRLA, V. "PRAVDĚPODOBNOST A STATISTIKA," [In Czech] (Probability and statistics) 2013. [Online]. Available: https://homen.vsb.cz/~oti73/cdpast1/KAP02/PRAV2.HTM. [Accessed on 4. 3. 2018].

[3] "Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining," [Online]. Available: http://www.tfidf.com/. [Accessed on 25. 12. 2017].

[4] LANDAUER, Thomas K.; FOLTZ, Peter W.; LAHAM, Darrell. An introduction to latent semantic analysis. Discourse processes, 1998, 25.2-3: 259-284.

[5] HÁJEK, Petr, et al. Možnosti využití přístupu indexování latentní sémantiky při předpovídání finančních krizí. POLITICKÁ EKONOMIE, [In Czech] (Possible use of indexed latent semantic approach for financial crisis prediction) 2009, 6: 755.

[6] "Support Vector Machines (SVM)," TIBCO Software Inc, [Online]. Available: http://www.statsoft.com/Textbook/Support-Vector-Machines. [Accessed on 28. 12. 2017].

[7] ŽIŽKA, J. "Studijní materiály předmětu FI:PA034," [In Czech] (Study materials to FI:PA034) [Online]. Available: https://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf. [Accessed on 29. 12. 2017].

[8] CAVNAR, William B., et al. N-gram-based text categorization. Ann arbor mi, 1994, 48113.2: 161-175.

[9] HABROVSKÁ, P. "Vybrané kapitoly z počítačového zpracování přirozeného jazyka," 2010. [In Czech] (Seleted chapters from natural language processing) [Online]. Available: http://www.inflow.cz/kratce-o-zpracovani-prirozeneho-jazyka.

[10] SCAGLIARINI, L.; VARONE, M. "Natural language processing and text mining," 11 April 2016. [Online]. Available: http://www.expertsystem.com/natural-language-processing-and-text-mining/. [Accessed on 15. 12. 2017].

[11] KODIMALA, Savitha. Study of stemming algorithms. 2010.

[12] RISUENO, T. "The difference between lemmatization and stemming," 28. 1. 2018. [Online]. Available: https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/. [Accessed on 4. 3. 2018].

[13] ŠMERK, P.; RYCHLÝ, P. "Majka – rychlý morfologický analyzátor," [In Czech] (Majka - quick morphological analyzer) 2009. [Online]. Available: https://www.muni.cz/vyzkum/publikace/935762. [Accessed on 15. 12. 2017].

[14] ESTELLÉS-AROLAS, Enrique; GONZÁLEZ-LADRÓN-DE-GUEVARA, Fernando. Towards an integrated crowdsourcing definition. Journal of Information science, 2012, 38.2: 189-200.

[15] SCHENK, Eric; GUITTARD, Claude. Crowdsourcing: What can be Outsourced to the Crowd, and Why. In: Workshop on Open Source Innovation, Strasbourg, France. 2009.

[16] AITAMURTO, Tanja; LEIPONEN, Aija; TEE, Richard. The promise of idea crowdsourcing–benefits, contexts, limitations. Nokia Ideasproject White Paper, 2011, 1: 1-30.

[17] KALSI, M. "Crowdsourcing through Knowledge Marketplace," 3. 3. 2009. [Online]. Available: http://blog.spinact.com/knowledge_as_a_service/2009/03/crowdsourcing-through-knowledge-marketplace-.html. [Accessed on 2018 3. 4.].

[18] KAUFMANN, Nicolas; SCHULZE, Thimo; VEIT, Daniel. More than fun and money. Worker Motivation in Crowdsourcing-A Study on Mechanical Turk. In: AMCIS. 2011. p. 1-11.

[19] VRL, NICTA. An unsupervised approach to domain-specific term extraction. In: Australasian Language Technology Association Workshop 2009. 2009. p. 94.

[20] TSAPATSOULIS, Nicolas; DJOUVAS, Constantinos. Feature extraction for tweet classification: Do the humans perform better? In: Proceedings of the 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP 2017), pp. 53-58, Bratislava, Slovakia, July 2017.

[21] TSAPATSOULIS, Nicolas; DJOUVAS, Constantinos. Opinion mining from social media short texts: Does collective intelligence beat deep learning? Frontiers in Robotics and AI, vol. 5, article 138, January 2019, DOI: 10.3389/frobt.2018.00138

[22] SUROWIECKI, James. The Wisdom of Crowds. 2005.

[23] KEARNS, K. "9 Great Examples of Crowdsourcing in the Age of Empowered Consumers," 10. 7. 2015. [Online]. Available: http://tweakyourbiz.com/marketing/2015/07/10/9-great-examples-crowdsourcing-age-empowered-consumers/. [Accessed on 10.3. 2018].