The Efficiency of Different Distance Metrics for Keyword-Based Search in Medical Documents: A Short Case Study

Ágnes VATHY-FOGARASSY[a,1], Szabolcs SZEKÉR[a], Balázs SZOLÁR[a] and György FOGARASSY[b]
[a] *Department of Computer Science and Systems Technology, University of Pannonia, Hungary*
[b] *State Hospital for Cardiology, Hungary*

Abstract. Background: Processing of free text written medical texts involves many difficulties arising from typographical errors, synonyms, and abbreviations occurring in the texts. Methods: In this study, the applicability of the most common string similarity measures were analyzed and compared for the keyword-based medical text search. Results: The usefulness of the similarity measures was studied in a set of medical documents containing more than 20,000 echocardiography reports. Experimental results showed that the Jaro-Winkler dissimilarity measure is the most capable measure to explore the content of the medical texts.

Keywords. recommendations, data annotation, word embedding, FastText

1. Introduction

Processing of Electric Medical Records (EMR) written in free text is a challenging task and requires different Natural Language Processing (NLP) techniques. Numerous international studies were also presented in this field [1-9], but the main problem is that a general and widely applicable solution does not exist. In practical solutions, direct search-based methods are the most generally applied methods to extract specific terms from free-text documents [1-3]. Furthermore, it is also typical, that only one parameter or a set of predefined parameters (e.g. chamber dimensions, flow velocity, etc.) is aimed to be extracted from the medical texts, like in [5], where NLP-based parsing and outlier analysis were utilized. Besides these direct approaches, pattern matching is also a popular method to extract terms from clinical notes [4, 9]. Furthermore, word embedding has also been successfully applied in the border-line field of the social and medical science to process tweets about healthcare-related events [10].

Our current research is a part of the development of a medical recommendation system capable of presenting relevant echocardiography reports to medical personnel based on given search terms and highlighting relevant parts of the document, making it easier to search through medical records. As exact matching is not capable of finding synonyms, typos, and abbreviations of the search term, the aim of our research is to

[1] Corresponding Author: Ágnes Vathy-Fogarassy, Department of Computer Science and Systems Technology, University of Pannonia, 2. Egyetem Str., 8200 Veszprém, Hungary, E-Mail: vathy@dcs.uni-pannon.hu

examine and compare different similarity measures applied in the field of NLP, and to determine which similarity measures present the highest gain in terms of searching for the documents containing the given keyword or its misspelled, abbreviated or synonym form.

2. Methods

The basic distance metrics included in our case study are widely used metrics in the field of NLP. The measures of the study were the following: Longest Common Subsequence (LCS), Levenshtein distance (LD), weighted Levenshtein distance (WLD), Jaro-Winkler distance, and cosine distance. In the following, these metrics and the principles behind them are introduced in detail.

2.1.1. Longest Common Subsequence

Longest Common Subsequence (LCS) is one of the simple metrics measuring the similarity of two strings. It finds the longest subsequence of characters present in both texts. To measure the similarity of the two strings, the actual common subsequence is irrelevant, only the length of it is taken into account [11]. For example, both 'cardi' and 'ardil' are subsequences of 'cardiology' and their length in both cases equals to five. The term subsequence is defined as follows. Given a sequence $x = x_1 \dots x_k$. Another $y = y_1 \dots y_m$ sequence is a subsequence of x if such a strictly increasing sequence of indices $(i_1, \dots, i_m)$ of x exists that for all $j = 1, \dots, m$, $x_{i_j} = y_j$. This metric also takes the cases into account where some characters are omitted, but it cannot recognize swapped characters. The Longest Common Subsequence metric was involved in our analysis, as it is a widely used common distance measure.

2.1.2. Levenshtein distance

The Levenshtein distance [12,13] is a more complex dissimilarity measure that counts the number of the edits that are needed to transform an x string into another y string. The Levenshtein distance takes the following operations into account: insertion, deletion, and substitution of characters.

The Levenshtein distance between strings x and y is denoted as $lev_{x,y}(|x|, |y|)$, where $|x|$ and $|y|$ yield the lengths of strings x and y, and $lev_{x,y}(i, j)$ for each $i, j \in \mathbb{N}$ is calculated as:

$$\mathrm{lev}_{x,y}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} \mathrm{lev}_{x,y}(i-1,j) + 1 \\ \mathrm{lev}_{x,y}(i,j-1) + 1 \\ \mathrm{lev}_{x,y}(i-1,j-1) + 1_{(x_i \neq y_j)} \end{cases} & \text{otherwise,} \end{cases} \quad (1)$$

where $1_{(x_i \neq y_j)}$ is equal to 0 if $x_i = y_j$, otherwise $1_{(x_i \neq y_j)}$ is equal to 1.

The inclusion of the Levenshtein distance in our analyses was motivated by the fact that in medical texts the Latin medical terms are probably written, to some degree, in a way similar to spoken-language, and this kind of difference between two words can be easily caught by the use of the Levenshtein distance. The Levenshtein distance works

basically on single words, however, it is not restricted to those: it can also be calculated for strings of any type.

2.1.3. Weighted Levenshtein distance

The original Levenshtein distance is not flexible enough to consider the magnitude of errors: all edit operations uniformly cost 1. However, practically, not all edits can be considered equivalent. For example, in case of typo correction substituting '*r*' for '*t*' should have a smaller cost, since they are located close to each other on a keyboard with QWERTY layout. The weighted Levenshtein distance considers all these aspects as well and sets different costs to the pairs of characters according to the probability of their interchange.

2.1.4. Jaro-Winkler distance

The Jaro-Winkler distance [13] accounts for the lengths of two strings and partially accounts for the type of typographical errors humans make when typing texts. It is calculated in the following way:

$$d_w\,(s_1, s_2) = 1 - sim_w\,(s_1, s_2) \tag{2}$$

$$sim_w\,(s_1, s_2) = sim_j\,(s_1, s_2) + lp(1 - sim_j\,(s_1, s_2)) \tag{3}$$

where sim_j is the Jaro similarity for s_1 and s_2 strings, l is the length of a maximum 4 characters long common prefix and p is a constant scaling factor with a standard value of 0.1. The Jaro similarity (sim_j) is calculated as:

$$sim_j(\mathrm{s}_1, \mathrm{s}_2) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise,} \end{cases} \tag{4}$$

where $|s_i|$ is the length of s_i, m is the number of matching characters and t is half of the number of transpositions. The concept of matching and transpositions is detailed in [13].

The Jaro-Winkler distance measure results in smaller distance values for those two strings that match from the beginning in length l. We decided to analyze the applicability of this kind of distance metric as well, because our hypothesis based on the manual review of a preselected sample document was that typing errors are more common toward the end of the words.

2.1.5. Cosine similarity

The cosine similarity is also a widely used similarity metric for comparing two strings. For example, it was used in anomaly detection in web documents [14], in content-based recommender systems [15], and even it was used for pattern recognition in medical diagnoses [16]. In the case of calculating cosine similarity, the strings are represented as vectors, and the similarity is calculated from the angle enclosed by the vectors. More formally, the cosine similarity is defined as the inner product of two vectors divided by the product of their lengths. To get the cosine similarity of two strings, the compared strings first have to be projected to an N-dimensional vector space. In our study, it has been done by applying word embedding.

Word embedding [17] is one of the most popular representations of document vocabulary as it is capable of capturing the context of a word in a document, semantic

and syntactic similarities or even relations between words. It provides an efficient representation in which similar words have similar encodings. As a result, the words that occur in a similar context will be represented as similar high-dimensional vectors and they tend to have high cosine similarity, as well.

In our study, the FastText word embedding library developed by Facebook AI Research (FAIR) team was used to calculate the high-dimensional vector representation for words occurring in medical texts. FastText is an extension of the Word2Vec model proposed by Google [17]. It uses a two-layer neural network for high-dimensional representation. The input of FastText is the word to be mapped with the surrounding text and the output is a high-dimensional (typically several hundred dimensions) representation of the word. The key difference between Word2Vec and FastText is the use of n-grams: Word2Vec only learns from complete words found in the training corpus, while FastText not only considers the complete words, but also the n-grams that are found within each word.

Having the high-dimensional representations of the strings to be compared, the cosine similarity is calculated as follows:

$$sim(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}, \tag{5}$$

where vectors $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$, and $\boldsymbol{y} = [y_1, y_2, \dots, y_n]$ are the N-dimensional vector representations of strings (in our case words) x and y to be compared.

The examination of cosine similarity was based on the fact that although the same condition may be formulated differently in medical descriptions, but the text surrounding the condition is most likely similar. Our hypothesis was, that with the use of this similarity measure, the different descriptions of the same medical terms (e.g. Latin and Hungarian forms of the same term) will get high cosine similarity value. As the similarity of the synonyms cannot be expressed by the application of similarity metrics presented before, we had great hope to use this measure.

2.2. Study design

While conducting the study, the study design presented in Figure 1 was applied. Our cohort contained 20074 Hungarian echocardiography reports. The number of unique expressions in the 20074 reports was 25380.

All echocardiography reports have been preprocessed as the zeroth step. The aim of preprocessing was to identify the measurements recorded as numerical values and to connect them to their units. As the different measured values were replaced with a unified special character, they did not differ in the preprocessed text, but later they can be mapped back to their original values. This way, the variability of the text resulting from different measurement results was significantly reduced, and the size of the dictionary also reduced to 15105.

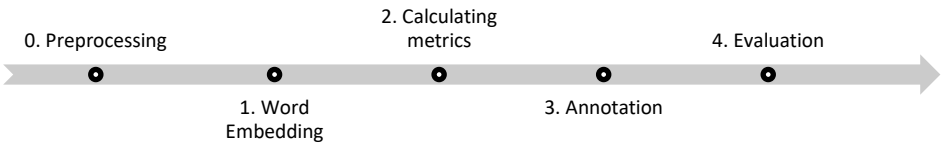

Figure 1. Workflow of the study

The next step of the work was the design and the execution of word embedding. The applied neural network was tested with different parameter settings, but in this paper, the settings belonging to the best results are only presented. The best performing neural net utilized a skipgram model with negative sampling. The number of negative samples was 5 and the sampling threshold applied was 10^{-4}. The learning rate was set to 0.05 with a rate of 100 for updating the learning rate. With 20 epochs and a window size of 5, all the words have been mapped to a 100-dimensional vector.

To evaluate the usefulness of the different similarity measures, 10 medical terms considered as important search keywords have been selected by the collaborating doctors, based on personal preferences. These terms were the following: *regurg*, *hypokinesia*, *akinesia*, *shunt*, *bicuspidal*, *thrombus*, *stenosis*, *systolic*, *mitral*, *wallmotion abnormality*. We have to note, that all these conditions were given according to the Hungarian terminology, where all of them are expressed with one word. Furthermore, we know that the set of the selected words is rather small, but in a later step, doctors have to manually annotate the resulting similar words. The difficulty of the task is shown by the fact that the evaluation of the similar words found for these 10 search words required annotations of 8647 similar words.

In the second step, the distance metrics have been calculated for the 1000 nearest matches of the previously presented keywords. For the final evaluation, all calculated distance values have been converted to similarity values and the converted values have been normalized to the range of $[0,1]$.

Our third step was the annotation. With the help of a cardiologist, a subset of the closest words has been annotated according to the following rules: (i) A word was labeled 2, if it was considered identical to the term searched for (e.g. alternative forms, abbreviations, typos). (ii) The word was labeled 1, if based on the found word, the cardiologist would consider checking the report containing it. (iii) Finally, a word was labeled 0, if the found word was considered irrelevant.

As the final step, the ROC curves have been plotted and the corresponding AUC (Area Under the Curve) values have been calculated based on the annotation labels for each search word: two values for each. The first called *hard evaluation* was calculated where only the words labeled 2 were considered matches, and the second one was the *soft evaluation*, where the words labeled as 1 and 2 were also considered relevant matches.

3. Results

The candidate words were evaluated based on their annotation labels. Following a data analysis step, we established that the Jaro-Winkler and cosine similarities resulted in more candidate words than the LCS, Levenshtein and weighted Levenshtein metrics. Table 1 shows the resulted number of candidate words in case of setting the similarity threshold equal to 0.65.

The corresponding AUC values for the results presented in Table 1 can be seen in Table 2. The notation '-' means that with this parameter setting, exclusively real positive candidates were selected and therefore the area under the ROC curve could not be calculated. As we can see, the LCS, Levenshtein, and weighted Levenshtein distances are more capable of distinguishing the true positive candidates from the false positive ones. The advantage of using the weighted version of the Levenshtein distance versus the basic one cannot be observed.

Table 1. The number of candidate words in case of applying different similarity measures and evaluation methods, setting the similarity threshold equal to 0.65. N_C yields the number of candidates, N_H and N_S the number of the true positive terms for the hard and soft evaluations respectively.

	Number of candidate words														
	LCS			**Levenshtein**			**weighted Lev.**			**Jaro-Winkler**			**Cosine**		
Term	N_C	N_H	N_S	N_C	N_H	N_S	N_C	N_H	N_S	N_C	N_H	N_S	N_C	N_H	N_S
regurg	17	15	15	15	15	15	15	15	15	103	25	29	155	54	91
hypokinesia	76	69	73	58	53	55	34	33	34	470	297	316	489	268	276
shunt	6	6	6	6	6	6	11	6	6	167	19	20	466	48	54
bicuspidal	6	2	2	6	2	2	5	2	2	135	3	4	1000	3	3
thrombus	25	25	25	22	22	22	23	23	23	191	41	49	97	42	52
stenosis	7	6	6	6	6	6	6	6	6	237	15	19	1000	15	52
systolic	50	19	39	45	19	34	39	17	27	259	27	83	136	30	89
akinesia	5	4	4	5	4	4	2	1	1	306	35	36	549	40	62
mitral	27	15	15	22	13	13	12	10	10	519	20	25	151	20	23
wallmotion abnormality	33	28	33	23	18	23	19	17	19	269	28	68	220	28	87

Table 2. AUC values in case of applying different similarity measures and evaluation methods, setting the similarity threshold equal to 0.65.

	AUC									
	LCS		**Levenshtein**		**weighted Lev.**		**Jaro-Winkler**		**Cosine**	
Term	**Hard**	**Soft**	**Hard**	**Soft**	**Hard**	**Soft**	**Hard**	**Soft**	**Hard**	**Soft**
regurg	1.0000	1.0000	-	-	-	-	0.9954	0.9646	0.8894	0.8398
hypokinesia	0.6004	0.7032	0.6038	0.8545	0.1818	-	0.8483	0.8609	0.8657	0.8695
shunt	-	-	-	-	0.9333	0.9333	0.9968	1.0000	0.9041	0.9143
bicuspidal	1.0000	1.0000	0.8750	0.8750	1.0000	1.0000	0.9975	0.9790	0.9997	0.9997
thrombus	-	-	-	-	-	-	0.9865	0.9789	0.9519	0.8850
stenosis	0.6667	0.6667	-	-	-	-	0.9688	0.9138	0.7919	0.7464
systolic	0.9440	0.7389	0.9413	0.8396	0.7701	0.7191	0.9120	0.9371	0.8041	0.5675
akinesia	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8741	0.8507	0.7622	0.6622
mitral	0.9778	0.9778	0.9829	0.9829	0.8000	0.8000	0.9704	0.9670	0.8729	0.8441
wallmotion abnormality	0.9286	-	0.7000	-	0.9118	-	0.8718	0.9606	0.9269	0.9243

Comparing the two similarity measures that selected more words as candidates, namely the Jaro-Winkler distance and the cosine distance, we can see that the application of the Jaro-Winkler distance is more appropriate in this topic. After manually reviewing the results, we have established, that although the cosine distance can extract completely different word synonyms as well, they appear in the result set with lower similarity values. Consequently, the false positive results appear with greater similarity values in the results set than the completely different forms of synonyms. Based on our findings, we have to reject the hypothesis that cosine similarity based on the applied FastText embedding can significantly improve keyword-based search in medical texts. The results can be explained by the fact that the contexts surrounding synonyms are probably different for those medical texts that use different expressions for the same content.

Our study showed, that for the LCS, Levenshtein, and weighted Levenshtein measures a lower, and for Jaro-Winkler and cosine similarity measures a higher threshold value has to be applied to get enough true positive candidate words in a way to get good enough AUC value for the classifier. For example, in Figure 2 the ROC curves for searching for the word '*thrombus*' is presented. For finding more positive samples the

similarity threshold for LCS, Levenshtein, and weighted Levenshtein distances was decreased to 0.5, while the threshold for Jaro-Winkler and cosine similarities was left equal to 0.65. The number of the true positive candidates in this case were: $N_{LCS} = 39$, $N_{Levenshtein} = 37$, $N_{wLevenshtein} = 37$, $N_{Jaro-Winkler} = 41$, $N_{Cosine} = 42$.

Comparing the results to data listed in Table 1 and Table 2, we can see, that as the threshold decreased for the first three similarity measures, the number of positive candidates increased, however the area under the ROC curve decreased.

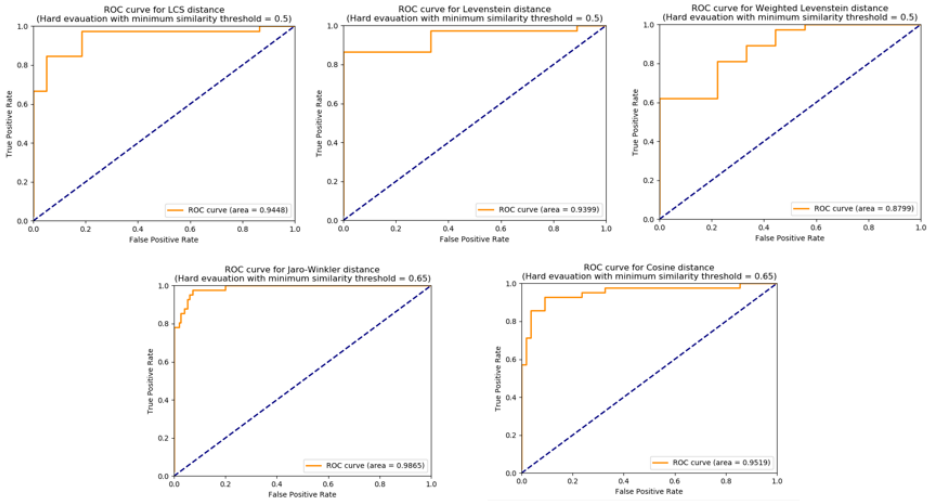

Figure 2. ROC analysis of finding the term '*thrombus*' using different similarity measures.

4. Discussion

In this study, our goal was to lay the foundations of a recommendation system that can found relevant medical documents based on a given search keyword. For this purpose, the applicability of different string similarity measures was tested and compared. The similarity measures involved in the study were: Longest Common Subsequence distance, Levenshtein distance, weighted Levenshtein distance, Jaro-Winkler distance, and cosine distance. Our experimental results show that the Jaro-Winkler and cosine distances at the same similarity threshold can discover more candidate words to be similar to the keyword. Although a classifier based on the Common Subsequence, Levenshtein, or weighted Levenshtein distances with higher similarity thresholds is more capable of distinguishing the true positive candidates from the false positive ones, with these high thresholds, these metrics provide less true positive results. Furthermore, our study pointed out, that the weighted Levenshtein distance cannot substantially contribute to improving the result of the Levenshtein distance. Considering the applicability of the cosine distance based on the FastText word embedding, we found that the exploration of synonyms requires a significantly lower threshold, which results in the decrease of the efficiency of the classifier, as well. In our study, the most promising distance measure was the Jaro-Winkler distance, which can return a relatively large number of documents in a way that the distinctive ability of the classifier still remains high. In our further research, we plan to combine the LCS, Levenshtein and Jaro-Winkler measures in a way that a relatively

high similarity threshold is applied to avoid the false negative results, besides to obtain high number of relevant matches. The search engine of the recommender system under development will apply the most effective simple or complex similarity measure with an optimized threshold value. Furthermore, we are looking for international research cooperation to compare our results arising from processing Hungarian medical documents with results of processing medical texts written in other languages.

Acknowledgement

We acknowledge the financial support of Széchenyi 2020 under EFOP-3.6.1-16-2016-00015 and UNKP-19-3 New National Excellence Program of the Ministry of Human Capacities, and gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

[1] F. Xie, C. Zheng, A. Yuh-Jer Shen, W. Chen, Extracting and analyzing ejection fraction values from electronic echocardiography reports in a large health maintenance organization, *Health Inform J.* **23**(4) (2017), 319–328.

[2] J.H. Garvin, S.L. DuVall, B.R. South, et al., Automated extraction of ejection fraction for quality measurement using regular expressions in Unstructured Information Management Architecture (UIMA) for heart failure, *Journal of the American Medical Informatics Association* **19**(5) (2012), 859–866.

[3] Y. Kim, J.H. Garvin, M.K. Goldstein, et al., Extraction of left ventricular ejection fraction information from various types of clinical reports, *Journal of Biomedical Informatics* **67** (2017), 42–48.

[4] O.V. Patterson, M.S. Freiberg, M. Skanderson, et al., Unlocking echocardiogram measurements for heart disease research through natural language processing, *BMC Cardiovascular Disorders* **17**(1) (2017), 151.

[5] Q.S. Wells, E. Farber-Eger, D.C. Crawford, Extraction of echocardiographic data from the electronic medical record is a rapid and efficient method for study of cardiac structure and function, *Journal of Clinical Bioinformatics* **4**(1) (2014), 12.

[6] M. Toepfer, H. Corovic, G. Fette, et al., Fine-grained information extraction from German transthoracic echocardiography reports, *BMC Medical Informatics and Decision Making* **15**(1) (2015), 91.

[7] S.R. Jonnalagadda, A.K. Adupa, R.P. Garg, et al., Text Mining of the Electronic Health Record: An Information Extraction Approach for Automated Identification and Subphenotyping of HFpEF Patients for Clinical Trials, *Journal of Cardiovascular Translational Research* **10**(3) (2017), 313–321.

[8] V. Renganathan, Text Mining in Biomedical Domain with Emphasis on Document Clustering, *Healthcare Informatics Research* **23**(3) (2017), 141–146.

[9] S. Szekér, G. Fogarassy, K. Machalik, Á. Vathy-Fogarassy, Application of Named Entity Recognition Methods to Extract Information from Echocardiography Reports, *Studies in Health Technology and Informatics*, **260** (2019), 41–48.

[10] X. Dai, M. Bikdash, B. Meyer, From social media to public health surveillance: Word embedding based clustering method for twitter classification, In SoutheastCon 2017, IEEE, (2017), 1–7.

[11] S. Lonardi, String Processing and Information Retrieval, Springer, 2010.

[12] F.P. Miller, A.F. Vandome, J. McBrewster, Levenshtein Distance: Information theory, Computer science, String (computer science), String metric, Damerau-Levenshtein distance, Spell checker, Hamming distance, Alpha Press, 2009.

[13] J. Piskorski, M. Sydow, String distance metrics for reference matching and search query correction, In International Conference on Business Information Systems (2007), 353–365.

[14] M. Friedman, M. Last, Y. Makover, A. Kandel, Anomaly detection in web documents using crisp and fuzzy-based cosine clustering methodology, *Information sciences* **177**(2) (2007), 467–475.

[15] A. Zenebe, A.F. Norcio, Representation. similarity measures and aggregation methods using fuzzy sets for content-based recommender systems, *Fuzzy sets and systems* **160**(1) (2009), 76–94.

[16] J. Ye, Cosine similarity measures for intuitionistic fuzzy sets and their applications, *Mathematical and computer modelling* **53**(1-2) (2011), 91–97.

[17] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, In Advances in neural information processing systems (2013), 3111–3119.