A Light-weight Text Summarizer for Fast Access to Medical Evidence

Abeed Sarker, PhD[1], Yuan-Chi Yang, PhD[1], Mohammed Ali Al-Garadi, PhD[1]
[1]Department of Biomedical Informatics, School of Medicine, Emory University, Atlanta GA 30033, U.S.A

Abstract

The performances of current medical text summarization systems rely on resource-heavy domain-specific knowledge sources, and preprocessing methods (e.g., classification or deep learning) for deriving semantic information. Consequently, these systems are often difficult to customize, extend or deploy in low-resource settings, and are operationally slow. We propose a fast summarization system that can aid practitioners at point-of-care, and, thus, improve evidence-based healthcare. At runtime, our system utilizes similarity measurements derived from pre-trained domain-specific word embeddings in addition to simple features, rather than clunky knowledge bases and resource-heavy preprocessing. Automatic evaluation on a public dataset for evidence-based medicine shows that our system's performance, despite the simple implementation, is statistically comparable with the state-of-the-art.

Introduction

Current clinical practice guidelines urge practitioners to follow the principles of evidence-based medicine, which requires them to integrate the best external scientific evidence with clinical expertise.[1,2] Early and recent studies show that one of the biggest obstacles to evidence-based medicine practice is information overload caused by the large volume of medical literature available.[3] Searching through medical evidence is too time consuming and practitioners often consider the task to be unproductive.[4] A standard clinical query on PubMed,[1] which indexes over 30 million articles, typically returns multiple pages of research publications. Hersh *et al.*[5] discussed that it takes more than 30 minutes for a practitioner to search for an evidence-based answer, and, particularly at point-of-care, practitioners cannot afford to spend that much time. Literature searching, and fast access to relevant information, is particularly beneficial for medical students and young practitioners because of their lack of clinical experience.

Motivated by the importance of incorporating scientific evidence in everyday medicine practice, automated text summarization research has attempted to address the problem. In particular, query-focused text summarization approaches have been explored to aid evidence-based medicine.[6–8] These systems take queries (in natural language or key-terms) as input and output query-relevant summaries. In terms of automatic summary qualities, the performances of successful approaches designed for the medical domain have relied heavily on domain-specific knowledge sources.[9] For example, one study Demner-Fushman and Lin[10] incorporated sentence-level knowledge in a supervised classification system trained to detect *outcome* sentences. Sarker *et al.*[8] and ShafieiBavani[7] utilized manually annotated summarization datasets specific to the domain to generate extractive and abstractive summaries—both systems relying heavily on the identification of domain-specific generalizations, concepts, and associations. Hristovski *et al.*[11] proposed the use of domain-specific semantic relations for performing question answering for biomedical literature. Further discussion of such systems is outside the scope of this short paper, and detailed descriptions of medical text summarization systems over the years are available through survey papers.[12,13]

Adaptation of summarization systems to a particular domain can be computationally expensive, and require large numbers of external tools.[14] Within the medical domain, systems typically use the Unified Medical Language System,[2] and tools such as MetaMap[15] are employed to derive domain-specific knowledge from lexical representations. This is in turn used in downstream tasks, or as features in learning systems. Heavy dependence on these domain-specific resources introduces disadvantages: (i) the systems are not very portable, (ii) they are difficult to re-implement and/or deploy without the knowledge sources, and (iii) the systems are slow and resource-heavy.

The goal of our work is to design a light-weight and fast medical text summarization system that is decoupled from knowledge bases. In our view, such a system may be operationalized as a web-based service, enabling practitioners to review medical evidence at point-of-care. The proposed system relies on publicly available labeled and unlabeled data, dense word vectors learned from the unlabeled data, and a set of simple features that require little

[1] https://www.ncbi.nlm.nih.gov/pubmed/ [Access date: 25th Nov 2019].
[2] http://www.nlm.nih.gov/research/umls/ [Access date: 12th Dec 2019].

computational resources and time. Automatic evaluation of our system with a state-of-the-art system on a standard dataset shows that our approach produces comparable summary quality.

Methods

We use a public dataset—the corpus provided by Molla-Aliod *et al.*,[16] which contains a total of 456 questions along with expert-authored single- and multi-document evidence-based summarized responses to them. Each question is generally associated with multiple single-document summaries, which present evidence from distinct studies. The abstracts of the studies from which the answers were derived are made available from PubMed. In total, the corpus contains 2,707 single-document summaries. To ensure fair comparison, we obtained the exact train-test split from the authors of the QSpec system[8] (1,388 for training, 1,319 for evaluation). The training set is used to devise feature scoring techniques and learn weights for all the feature scores. The summary sentences are scored as the sum of the weighted feature scores, and chosen sequentially (from first to last), taking into consideration the target sentence position and the contents of the chosen sentences. The scoring process can be summarized as: $\boldsymbol{S_{m,t_n}} = \sum_{i=1}^{k}(\boldsymbol{w_i} \times \boldsymbol{f_i}|\boldsymbol{t_n})$, where S_{m,t_n} is the score for sentence number m of a text, given the summary target sentence number t_n, and w_i and f_i are the weight and score for feature i, respectively. For each t_n, the top-scoring sentence is chosen. Following experimentation with multiple features, we selected five that could be computed fast, and proved to be useful when used in combination. We describe these in the following paragraphs.

Word Embedding-based Maximal Marginal Relevance

Maximal Marginal Relevance (MMR)[17] is a strategy that can be used to increase relevance and reduce redundancy in summarization. The core of the technique relies on computing two similarity measures—between sentences and the associated query, and between the sentences themselves. During score generation, sentences are rewarded for being similar to the query, while at the same time, they are penalized for being similar to sentences that have already been chosen to be included in the summary. The similarity values are combined linearly with suitable weights ($\boldsymbol{\lambda}$): $\boldsymbol{MMR = \lambda \times SIM(S_m, Q) - (1 - \lambda) \times max_{S_c \epsilon S_{sel}}\left(SIM(S_m, S_c)\right)}$, where $\boldsymbol{SIM(S_m, Q)}$ is the similarity score between a sentence and the question and $\boldsymbol{max_{S_c \epsilon S_{sel}}\left(SIM(S_m, S_c)\right)}$ is the maximum similarity between the same sentence and the set of already chosen summary sentences. Choosing the best three-sentence summary is a combinatorial optimization problem, and MMR enables us to approach sentence selection in a sequential manner.

We experimented with two variants of MMR that rely on the distributed representations of the words in the sentences and the questions. We obtained pre-trained embeddings that were generated from all PubMed and PMC OA texts[18] using the *word2vec* tool[3] (vector = 200, window size = 5) and the *skip-gram* model.[19] For the first variant, we compute the similarity between two text segments (*i.e.*, sentence *vs.* question and sentence *vs.* sentence) as the *average* cosine similarity of all the terms. We compute this average by adding the cosine similarities of all the term combinations and dividing by the product of the lengths of the two texts. For the second variant, we use the word vectors in a text segment to compute its centroid in vector space. A single centroid is computed for the set of all words within the set of already chosen sentences **($\boldsymbol{S_{sel}}$)** These centroids are then used to compute MMR.

Traditional MMR Score

For the traditional MMR score, we first preprocessed the terms by lowercasing, stemming and removing stop words. We then computed the *tf* × *isf* for each word in a sentence and the question—where *tf* is the frequency of a term in a text segment and *isf* is the *inverse sentence frequency* of the term in all the texts (*i.e.*, the inverse of how many sentences including the question contain the term). We then generate vectors for each sentence using the *tf* ×*isf* values of the terms.

Sentence Length Score

Sentence length is a metric that may filter out uninformative, short sentences by assigning them a lower score, while rewarding sentences that are relatively longer in a document. In past research in this domain, it has been mentioned to be particularly effective for tie-breaking. In our work, we attempted to assign penalties to very short sentences (*e.g.*, 1—3 word sentences), which often represent section headers. At the same time, our goal was to assign higher scores to longer sentences--- with decreasing gradients for very long sentences, such that this score does not play a significant role in choosing between those informative sentences.

[3] https://code.google.com/p/word2vec/. [Access date: 17th Nov 2019].

Our experiments on the training set suggested that a *sin* function conveniently served this purpose. The average sentence length in the training data is approximately 150 characters, so we considered 0 and 300 characters to be the lower and upper length limits, respectively, and mapped the lengths to the range $[-\pi/2, \pi/2]$. Following that, we applied a *sin* function to the mapped value to generate a length score between [-1,1] (Figure 1).

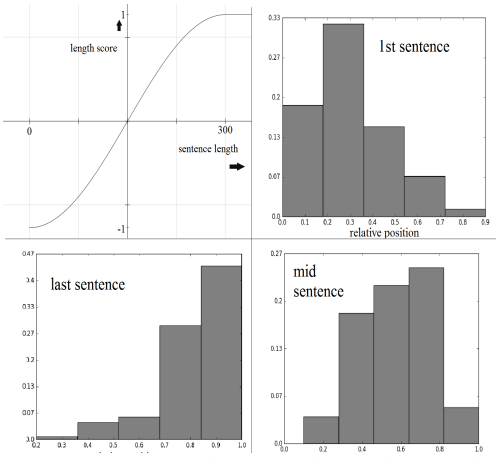

Figure 1 Clockwise from top-left: sine function for sentence length score, maxed at 300 characters; first, middle and last sentence relative position distributions from the best-scoring extractive training set summaries.

Sentence Position Score

Our last score is based on sentence position and the target sentence number. Sentence position has been shown to be a crucial metric for extractive summarization in domains including news[20] and medical.[10] We use an identical scoring approach to Sarker *et al.*[8] and apply target sentence specific summarization. We first obtain the best three-sentence summary for each training text, and use these sentences to generate normalized frequency distributions of the relative sentence positions for these *best* sentences for each of the three target sentence positions. During summary generation, given the relative sentence position *r* of a source sentence, the score assigned is the normalized frequency for *r* in the given target sentence distribution.

Weight Optimization and Sentence Scoring

We compute optimal weights for scoring using the training set via a grid search in the range [0.0,1.0] with step sizes of 0.1. For each weight combination, all the three-sentence training set summaries are generated and the ROUGE[21] summary evaluation tool is used to compare the extractive summaries with the expert-authored summaries in the corpus. The weights producing the highest F_1-score for the training set are used for evaluation on the test set.

Evaluation, Results and Discussion

As mentioned, we used the ROUGE summary evaluation tool to compare the performance of our system with other systems. Since the summaries are not length-restricted, we used the ROUGE-L F_1-score as our evaluation metric. ROUGE scores have been shown to be highly correlated with human evaluators.[21] Table 1 presents the performance of our system along with several other systems, including the state-of-the-art for this task (QSpec). Identical training-test splits were used for evaluation. The table shows that despite the simplicity of our approach, its performance is comparable to the state-of-the-art, and significantly better than other baselines.

Due to the simplicity of our approach, it can be easily re-implemented, customized or extended for real-life settings, and the results can be reproduced without requiring the use of third-party tools. Compared to the resource-heavy QSpec system, which requires query and sentence classification, and the generation of UMLS semantic types and associations, our approach requires minimal preprocessing. Only a set of pre-trained word embeddings are required. The light-weight nature of the summarizer also means that it runs faster than QSpec. On a standard computer (Intel® i5 2.0 GHz processor), it takes our summarizer a few minutes to summarize all the documents in the test set.

We obtained the word embeddings from past research and used them without modification. There is a possibility that the learning of these embeddings can be customized to the summarization task for improving performance. Furthermore, since the embeddings can be learned from publicly available data, and no other resources are required by the summarizer, the system can be implemented anywhere with minimal effort.

Table 1. Comparison of ROUGE-L F_1-scores for our summarizer with other systems and 95% confidence intervals.

System	ROUGE-L F_1 Score	95% CI
Our system	0.166	0.162 – 0.170
Sarker et al.[8]	0.168	0.164 – 0.172
Last 3 Sentences	0.155	0.151 – 0.158
Demner-Fushman and Lin[10]	0.159	0.152 – 0.164
Random	0.154	0.150 – 0.157
First 3 Sentences	0.140	0.136 – 0.143

Conclusions and Future Work

In this paper we presented a simple, query-focused, extractive summarization system suited for application in the domain of evidence-based medicine. Unlike past systems in this domain that rely heavily on domain-specific resources such as knowledge bases, our system is light-weight in nature and relies only on distributed representations of words learned from unlabeled text. Using a set of similarity-based and structural features, our system performs comparably to the state-of-the-art system, with a ROUGE-L F_1-score of 0.166. Because of the simplicity of our system, it can easily be ported and implemented in different settings.

References

1. Sackett DL, Rosenberg WMC, Gray JAM, Haynes RB, Richardson WS. Evidence based medicine: what it is and what it isn't. *BMJ*. 1996;312(7023):71-72. doi:10.1136/bmj.312.7023.71
2. Greenhalgh T. *How to Read a Paper : The Basics of Evidence-Based Medicine*. 5th ed. BMJ Books; 2014.
3. Ely JW, Osheroff JA, Ebell MH, et al. Analysis of questions asked by family doctors regarding patient care. *Br Med J*. 1999;319(7206):358-361. doi:10.1136/bmj.319.7206.358
4. Swennen MHJ, Van Der Heijden GJMG, Boeije HR, et al. Doctors' perceptions and use of evidence-based medicine: A systematic review and thematic synthesis of qualitative studies. *Acad Med*. 2013;88(9):1384-1396. doi:10.1097/ACM.0b013e31829ed3cc
5. Hersh WR, Katherine Crabtree M, Hickam DH, et al. Factors associated with success in searching MEDLINE and applying evidence to answer clinical questions. *J Am Med Informatics Assoc*. 2002;9(3):283-293. doi:10.1197/jamia.M0996
6. Cao YG, Liu F, Simpson P, et al. AskHERMES: An online question answering system for complex clinical questions. *J Biomed Inform*. 2011;44(2):277-288. doi:10.1016/j.jbi.2011.01.004
7. Shafieibavani E, Ebrahimi M, Wong R, Chen F. *Appraising UMLS Coverage for Summarizing Medical Evidence*. http://biotext.berkeley.edu/software.html. Accessed December 8, 2019.
8. Sarker A, Mollá D, Paris C. Query-oriented evidence extraction to support evidence-based medicine practice. *J Biomed Inform*. 2016;59. doi:10.1016/j.jbi.2015.11.010
9. Plaza L. Comparing different knowledge sources for the automatic summarization of biomedical literature. *J Biomed Inform*. 2014;52:319-328. doi:10.1016/j.jbi.2014.07.014
10. Demner-Fushman D, Lin J. Answering clinical questions with knowledge-based and statistical techniques. *Comput Linguist*. 2007;33(1):63-103. doi:10.1162/coli.2007.33.1.63
11. Hristovski D, Dinevski D, Kastrin A, Rindflesch TC. Biomedical question answering using semantic relations. *BMC Bioinformatics*. 2015;16(1):6. doi:10.1186/s12859-014-0365-3
12. Mishra R, Bian J, Fiszman M, et al. Text summarization in the biomedical domain: A systematic review of recent research. *J Biomed Inform*. 2014;52:457-467. doi:10.1016/j.jbi.2014.06.009
13. Athenikos SJ, Han H. Biomedical question answering: A survey. *Comput Methods Programs Biomed*. 2010;99(1):1-24. doi:10.1016/j.cmpb.2009.10.003
14. Severyn A, Moschittiy A. Learning to rank short text pairs with convolutional deep neural networks. In: *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Inc; 2015:373-382. doi:10.1145/2766462.2767738
15. Aronson AR, Lang FM. An overview of MetaMap: Historical perspective and recent advances. *J Am Med Informatics Assoc*. 2010;17(3):229-236. doi:10.1136/jamia.2009.002733
16. Mollá D, Santiago-Martínez ME, Sarker A, Paris C. A corpus for research in text processing for evidence based medicine. *Lang Resour Eval*. 2016;50(4). doi:10.1007/s10579-015-9327-2
17. Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '98*. New York, New York, USA: ACM Press; 1998:335-336. doi:10.1145/290941.291025
18. Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. *Distributional Semantics Resources for Biomedical Text Processing*. https://github.com/spyysalo/nxml2txt. Accessed December 8, 2019.
19. Mikolov T, Chen K, Corrado G, Dean J. Distributed Representations of Words and Phrases and their Compositionality. *Nips*. 2013:1-9. doi:10.1162/jmlr.2003.3.4-5.951
20. Barzilay R, Mckeown KR. *Sentence Fusion for Multidocument News Summarization*.; 2005.
21. Lin C-Y. *ROUGE: A Package for Automatic Evaluation of Summaries*.