Summarizing Citation Contexts of Scientific Publications

Sandra Mitrović[1], Henning Müller[1]

[1] University of Applied Sciences Western Switzerland (HES–SO)

Abstract. As the number of publications is increasing rapidly, it becomes increasingly difficult for researchers to find existing scientific papers most relevant for their work, even when the domain is limited. To overcome this, it is common to use paper summarization techniques in specific domains. In difference to approaches that exploit the paper content itself, in this paper we perform summarization of the citation context of a paper. For this, we adjust and apply existing summarization techniques and we come up with a hybrid method, based on clustering and latent semantic analysis. We apply this on medical informatics publications and compare performance of methods that outscore other techniques on a standard database. Summarization of the citation context can be complementary to full text summarization, particularly to find candidate papers. The reached performance seems good for routine use even though it was only tested on a small database.

Keywords: Text summarization, sentence similarity, citation context

1 Introduction

The increasing volume of produced research papers makes their use difficult and time–consuming, even for a small scientific domain. One way to quickly grasp the main results of a set of existing papers is through paper summarization. However, since publications can be long, this approach is not always efficient to get the most important aspects of a paper. Instead, the context in which a paper is cited can be used as indicator for its main contributions [6]. This context (known as *citation context* or *citation summary*) refers to a set of sentences pointing to the paper [17] when cited. If the publication is frequently cited, its citation context is also long, so we propose summarizing citation contexts longer than two sentences (otherwise, we consider them concise). We opt for generic extractive summarization where the aim is to extract original sentences that preserve the substance of the original text, leaving out potentially irrelevant details. In order to complete our task, we combine several existing approaches into a novel workflow and apply it on 50 randomly selected publications from our research group. We first extract and segment references, detect citations and merge them into integrated citation contexts. Then, we summarize them using methods based on clustering and latent semantic analysis (LSA). We did not restrict summary length in advance, since previous work suggests that such an

approach can affect summarization results [8]. Instead, we use an approach based on word distribution [1]. We compare algorithm performance using standard evaluation measures such as ROUGE [10] and the F1–measure on two data sets. We also explain challenges faced during different phases of our work including the small size of a citation context and relaxed grammatical structures, which increase the complexity of summarization.

2 Related Work

Reference extraction and segmentation Reference formats are not standardized. Hence, despite much existing work, there are continuous efforts to improve reference extraction. ParsCit[1] (successfully used in [12]) is the current state–of–the–art reference extraction system that uses both heuristics and conditional random fields. Another freely available tool for extracting metadata from scientific publications is PDFmeat[2], based on Google Scholar[3]. On the other side, efficient results were obtained even with regular expressions and heuristics [2].

Citation context extraction Identifying the full span of a citation context within a publication is a challenge. While previous work [4] suggests using a fixed character–length window around a citation, [19] concluded that sentence–based contexts are more effective than windows of equivalent length.

Text summarization During almost half a century, text summarization evolved into different branches. We constrain this overview to generic extractive single–document techniques. Generic means that summary refers to the main topic of the entire text. Extractive means that the parts of text conveying essential information are simply extracted without modification. A significant amount of work on extractive summaries uses statistical [24] and machine learning approaches [5, 22]. One of the most recent approaches is based on prior sentence clustering [16, 1], selecting for the summary the most representative sentences from each cluster. Another group of articles applies LSA [21, 15]. Text summarization is a challenging task due to anaphors and cataphors. Moreover, extractive summaries usually require human intervention to smooth the transition from one topic to another.

Sentence similarity Text clustering relies on sentence similarity to distinguish the most relevant parts of the document. Since citation contexts are usually short, we aim at determining sentence semantic similarity which reduces to word semantic similarity. The latter can be ontology/thesaurus–based or information theory/corpus–based (also called distributional) [9]. Onthology–based measures relate to the distance between concepts in onthology (known as path similarity) or to information content (e.g. [18]). Pointwise mutual information and LSA are two well–known techniques used in corpus–based similarity. The choice of the sentence similarity measure influences the summarization result [16, 1].

[1] `http://aye.comp.nus.edu.sg/parsCit/`
[2] `http://dbs.uni-leipzig.de/pdfmeatdemo/demo.html`
[3] `http://scholar.google.com/`

Evaluation of summaries We focus on direct (*intrinsic*) evaluation of summaries, where a summary is compared with a gold standard. Although it is not easy to agree on a gold standard, if it is available, the standard F1–measure can be used, as well as ROUGE [10], a widely accepted measure introduced at DUC[4]. ROUGE is based on statistical overlapping of gold standard and automatically created summary. The pyramid method [13] is a semi–automatic content–based method based on construction of pyramid containing so–called summarization content units. Methods without manual summaries appeared recently but the results obtained correlate well with ROUGE [23].

3 Materials and Methods

3.1 Data and Tools

We use 50 randomly selected publications belonging to researchers of the eHealth unit of HES–SO[5]. All publications are provided in PDF format (in English) and refer to medical information retrieval but differ in size and layout. We refer to this data set (and data extracted from it) as the HES–SO data set. Additionally, a benchmark called DUC2002 with 567 document–summary pairs was used for summarization evaluation (used as baseline in [1]).

Except for the Java library PDFBox[6], used to convert PDF to text, the code was entirely developed using Python NLTK [3] and the Scikit libraries. For storing all data we use MySQL. Summarization was implemented and run on a Hadoop[7] distributed computing platform. In our setting, map was performing summarization related calculations, while the reducer was responsible for storing summaries at the requested location in the database. In this manner, the reducer remains the same for different summarization methods.

3.2 Suggested Approach

Reference Extraction, Segmentation and Matching To precipitate pre–processing, we tried applying ParsCit and PDFmeat on the HES–SO data but both provided unsatisfactory results (on a paper with 52 references, ParsCit correctly extracted only the first 19, while PDFmeat substituted all authors' names (except the first) with "et al."). Thus, we decided to implement this part ourselves as a mixture of regular expressions and heuristics since we had no manually–annotated training set. Moreover, these fairly simple methods proved efficient [2].

Extracting references For identifying the reference section, apart from common starting keywords (as in [2, 12]), we had to include additional checks regarding section ends since 14% of the selected HES–SO papers had additional content

[4] Document Understanding Conference; `http://duc.nist.gov/`
[5] `http://medgift.hevs.ch/`
[6] `http://pdfbox.apache.org/`
[7] `http://hadoop.apache.org`

behind references (e.g. correspondence addresses). Next, we constructed regular expressions capturing numbered references (e.g. *[1].* or *1.*) since only these appeared in the sample data and drastically outnumbered non–numbered references in the complete publication set. The HES–SO data contained 1055 references, thus on average each paper had 21,1 references (min 6, max 61).

Reference segmentation We extract from each reference: author names, title, year, journal/venue, volume, number and pages (where applicable). As mentioned, reference formats are not standardized, differing in content, order of mentioned elements, separators used. We used four pattern types to capture the most dominant patterns of author names (see Table 1). To avoid overlaps and

Table 1. Pattern types used for capturing author names

Pattern type description	Examples
initials followed by surname	A. García Seco de Herrera; D. M. Van De Ville; L.-T. Guo; M.-A. Keller-Rex; G. McLeman; C. E. Kahn Jr.
surname followed by name or name initials	Van De Ville Dimitri; van Ginneken BJ; McLennan Geoffrey; da Costa JC; Shyu Chi-Ren; Leh TM (also Leh T M); Similowski Thomas
surname, initials	Fillion-Robin, J.-C.; Mazzoncini de Azevedo-Marques, P.; van Ginneken, B.; Guo, L.-T.; McLeman, G.; Bakke, B., Jr.; Leh, T.M.;
name surname	Jayashree Kalpathy-Cramer; Dimitri Van De Ville; Bruno van Ginneken; Lao-Tze Guo; Yasin Ben Salem

incorrect matches (e.g. "John Doe" matches both pattern 2 and 4), we developed four pre–parsers (one per pattern). While this handled situations where among several different author formats in a paper one was predominant, we still encountered a few exceptions: different author formats appearing within the same reference (e.g. *Thomas M. Deserno, Sameer Antani, and L. Rodney Long*), missing authors, typos etc. For extracting titles, we used NLTK [3] sentence extraction, working well except when title contained a dot sign or consisted of more than one affirmative sentence. For years, we modified 4 digit patterns to cover different date formats (e.g. *May/Jun 2012*), usually scanning reference string backwards. For volume, number and pages, we combined their dedicated patterns (e.g.: *vol. 3* or *p. 12-16*) with those allowing their common retrieval (e.g. *20(May(3)):26-39* or *75(1-2): 11-9*). Finally, the remainder of the reference was taken as journal/venue.

Reference matching As the same publication can be cited in different papers and using various formats (see Figure 1), it was essential to identify all the re–occurrences of the same paper in order to properly define citation context. We implemented 4 matching scenarios ranging from exact matching to similarity es-

timation based on heuristically determining similarity thresholds and Damerau-Levenshtein distance, modified to tolerate reasonable differences between two strings. These allowed matching when a list of authors is replaced with "et al.", when one author is accidentally omitted or when differences stem from special character misspellings (e.g. "Müller" (correct) vs. "Muller"/"Mueller") etc.

Müller, H., Michoux, N., Bandon, D., Geissbuhler, A.: A review of content-based image retrieval systems in medicine-clinical benefits and future directions. Int. J. Med. Inf. **73**(1) (2004) 1-23

H. Müller, N. Michoux, D. Bandon and A. Geissbuhler, "A review of content-based image retrieval systems in medicine-clinical benefits and future directions," *Internation Journal of Medical Informatics*, vol. 73, no. 1, pp. 1-23, 2004.

Fig. 1. Two formats of the same reference

3.3 Text Summarization

We perform text summarization using two approaches: clustering and LSA.

Similarity Measures for the Clustering-based Approaches We used two types of similarity measures for clustering: one based on a thesaurus referred to as **combined** and other, referred to as **distributional**.

1. Thesaurus–based similarity measures This similarity measure is an adaptation of the similarity measure used in [16] and represents a linear combination of three similarity measures. With all of them, for each particular citation context we dynamically create vocabularies eliminating stop words using the Python NLTK library. Then, each sentence is considered a bag of words.

For the first measure, the similarity between two sentences was calculated in the same way as in [16]: $sim_1(S_1, S_2) = \frac{2 * matched(S_1, S_2)}{num\_words(S_1) + num\_words(S_2)}$, where $matched(S_1, S_2)$ is the number of words that the two sentences S_1 and S_2 share and $num\_words(S)$ is the number of words that sentence S contains.

The second similarity measure in [16] was based on TF–IDF scores using uni–grams, bi–grams and tri–grams. We took into account only uni–grams, since cocitation formulations usually differ significantly [6]. The similarity between two sentences $sim_2(S_1, S_2)$ is calculated as cosine similarity of the corresponding sentences' TF–IDF vectors (more precisely, TF–ISF as in our setting, sentence corresponds to document, word to term and citation context to corpus).

It is worth noting that despite their similarities, first and second similarity measures express different concepts: while the first focuses exclusively on vocabulary overlap, the second emphasizes the overlapping word importance. The third similarity measure focuses on semantic similarity. Since we did not deal

with Chinese, instead of using HowNet[8] (as in [16]), we decided to use WordNet[9] [11], an enormous lexical database and online thesaurus in English. In WordNet, similarity is defined on the level of synsets (sets of near synonyms that share a common meaning (sense)). Thus, we define word–word similarity as maximal similarity between any two of their senses [9]:

$ww\_sim(w_1, w_2) = max\{ss\_sim(s_1, s_2) : s_1 \in synset(w_1), s_2 \in synset(w_2)\}$, where ss_sim is similarity between two senses s_1, s_2 (calculated using provided Python NLTK functions). Further, we define word–sentence similarity as in [16]: $ws\_sim(w, S) = max\{ww\_sim(w, v) : v \in S, v\ word\}$ and finally, sentence-sentence similarity as:

$sim_3(S_1, S_2) = \frac{\sum_{w_i \in S_1} ws\_sim(w_i, S_2) + \sum_{w_j \in S_2} ws\_sim(w_j, S_1)}{num\_words(S_1) + num\_words(S_2)}$

The final similarity measure is obtained as a linear combination of the three calculated measures: $sim(S_1, S_2) = \sum_{i=1}^{3} \lambda_i sim_i(S_1, S_2)$. We repeat the entire procedure twice: first, setting ss_sim in sim_3 to be a path similarity measure (obtaining thus similarity measure sim that we refer to as COMB_PATH), and second, using the Resnik similarity measure for ss_sim in sim_3 (denoting final similarity sim as COMB_RES). It is also worth mentioning that we use a general–purpose corpus *wordnet_ic* for generating an information content file applied to calculate the Resnik similarity.

Initially, we borrowed values of parameters λ from [16], since they also gave more importance to semantic similarity, but we also performed a small experiment varying the values (while retaining $\lambda_1 + \lambda_2 + \lambda_3 = 1$).

2. Distributional similarity measures In distributional algorithms words are similar if they have similar distributional contexts [9]. They are used to overcome the problems of a missing or incomplete thesauri. In this approach, we construct a word–context matrix which is based on positive pointwise mutual information (PPMI) [14], calculated as:

$$PPMI(w, c) = \begin{cases} PMI(w, c) = log_2 \frac{freq(w, c)}{freq(w) * freq(c)} : \ if\ PMI(w, c) > 0 \\ 0 : otherwise \end{cases}$$

where $freq(w, c)$ is the number of times that word w has context c, $freq(w)$ is the number of word w occurrences, $freq(c)$ is the number of context c occurrences. We build a PPMI matrix (with words as rows and contexts as columns) taking 20 words around the word as its context (to avoid computational complexity), apply add–one smoothing (to avoid bias toward infrequent occurrences) and define a word–word similarity measure, using Dice: $sim_{Dice}(v, w) = \frac{2 * \sum_i min(v_i, w_i)}{\sum_i (v_i + w_i)}$ and Jaccard similarity: $sim_{Jaccard}(v, w) = \frac{\sum_i min(v_i, w_i)}{\sum_i max(v_i, w_i)}$, where v_i is PPMI value for word v in the context i and w_i is PPMI value for word w in the context i. These two measures are selected as they perform better than cosine [20]. We then calculate similarity between sentences as: $sim(S_1, S_2) = \frac{\sum_{w_1 \in S_1} \sum_{w_2 \in S_2} word\_sim(w_1, w_2)}{\sqrt{num\_words(S_1) * num\_words(S_2)}}$, where $word\_sim$ is once Dice (denoted in fur-

[8] `http://www.keenage.com/`

[9] `http://wordnet.princeton.edu/`

ther text as PPMI_DICE) and another time Jaccard similarity (denoted as PPMI_JACCARD).

Clustering–based Approach Since we did not have training data, we experimented with three clustering methods: K-means, hierarchical agglomerative clustering (HAC) and affinity propagation (AP). For each of these, the four similarity measures were used. Due to different syntactic and semantic features of citation contexts, the number of clusters was not defined in advance. Instead, we calculated it based on the distribution of words [1] in the sentences of particular citation contexts: $K = n * \frac{|C|}{\sum_{i=1}^{n} |S_i|}$, where $|C|$ and $|S_i|$ are the number of words in citation context C and i–th sentence of citation context C respectively, n is the number of sentences in citation context C. Details can be seen in [1].

With K–means, we randomly selected K sentences for the K initial cluster centroids (Forgy method). A convergence to a global optimum with K–means cannot be guaranteed. Thus, to avoid obtaining clusters not reflecting the real situation, we ran the algorithm 10 times with random initializations and selected as final clustering the one with minimal intercluster similarity and maximal intracluster similarity.

In HAC, we followed the "bottom–up" approach, starting from clusters containing only one sentence and progressively merging them into bigger clusters. Among the three most popular linkage criteria determining how the distance/similarity between clusters can be calculated, we decided to apply average linkage clustering which defines linkage between two sets A and B as:

$\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$, where d is dissimilarity or similarity measure.

AP is a clustering method based on message passing between data points in the initial data set [7]. Unlike other clustering methods, for AP we used a method from the Python Scikit library. We kept all default parameters, except for three. First, we determined the number of clusters in the same way as for previous clustering algorithms; second, we used the explained similarity measures instead of the default (negative Euclidean) and third, we increased the number of iterations untill convergence from a default 15 to 20. Since we did not use the default affinity, the obtained result contained only cluster labels so additional coding was done to determine centroids.

LSA–based Approach LSA discovers latent semantic interrelationships among words, which allows identifying independent concepts hidden in the text. It applies singular value decomposition (SVD), factorizing a term–document matrix A (in our case word–sentence matrix) into a product of three matrices $U\Sigma V^T$. Σ is a diagonal matrix where non–zero entries are singular values, representing concepts. The magnitude of a singular value reflects the importance of the appropriate concept. The matrix V^T, with concepts as rows and sentences as columns, describes how important each concept for each sentence is, allowing capturing the most informative sentences.

Here, we implement two methods based on LSA.

CROSS method This method (introduced in [15]) is actually the modification of the Steinberger and Jezek [21] method and it often performs better than other LSA methods. In [21] sentence selection is based on sentence length, calculated as: $len(s_i) = \sqrt{\sum_{j=1}^{k} \sigma(j,j)^2 * v(j,i)^2}$, where s_i is i-th sentence, $v(j,i)$ is element of matrix V^T corresponding to the j–th concept and i–th sentence, and $\sigma(j,j)$ is singular value for j–th concept. The novelty in [15] (compared to [21]) is that the additional preprocessing step for matrix V^T is introduced in order to eliminate underrepresented sentences (where scores per concept are lower than the average sentence score per concept). Then, only the K longest sentences are taken for the summary, calculating K in the same way as with clustering methods.

HYBRID method As a second method we are proposing an approach where only a subset of singular values is taken into consideration based on the amount of information that we want to retain. After selecting the top X singular values to keep we calculate the strength of each sentence as: $strength(s_i) = \sum_{j \in sel\_concept} v(j,i)$, where $v(j,i)$ is corresponds to the j–th concept and i–th sentence in V^T. In the end, we select for the summary the top K strongest sentences, choosing K the same way as in previous methods.
We applied both methods on three types of word–sentence matrices:

1. *binary* matrix with $b_{ij} = 1$ if word i appears in sentence j and 0 otherwise
2. *root* matrix with values $r_{ij} = 1$ if word i appears in sentence j and word i is a noun, and 0 otherwise
3. *TF-ISF* matrix with values t_{ij} which represent TF-ISF score of word i with respect to the sentence j

4 Experimental Results

The accuracy of the reference extraction was 82% (of 1055 references to extract). It was evaluated by manually scanning original references and extracted information, considering that a reference is successfully processed only if all relevant data are correctly extracted.

For the citation extraction, we again manually checked the quality of the extracted citations. We obtained an accuracy of 83.5%. Actually, all extracted data contain valid citation sentences but we were not always able to exclude unnecessary (sub)titles, footers/headers, tables. Additionally, even though they were technically correctly extracted, citations obtained from table cells were considered as incorrect, due to their lack of context.

After matching, 885 unique papers remained, out of which 786 papers where cited only once, while 31 paper had more than 2 citations (1 paper had 16 citations, the maximum). We consider only these 31 papers for the summarization task. An example of the obtained summary compared with manually made one and original citation context can be seen in Table 2. We evaluate summaries

Table 2. Citation context, corresponding manual and generated summaries

citation context	manual summary	automatic summary
ImageCLEFmed is part of ImageCLEF focusing on medical images. 1 Introduction A medical retrieval task has been part of ImageCLEF1 since 2004. 1 Introduction ImageCLEF1 started in 2003 as part of the Cross Language Evaluation Forum	A medical retrieval task has been part of ImageCLEF1 since 2004. ImageCLEF1 started in 2003 as part of the Cross Language Evaluation Forum	1 Introduction A medical retrieval task has been part of ImageCLEF1 since 2004.ImageCLEFmed is part of ImageCLEF focusing on medical images

using the ROUGE–2 measure:

$$ROUGE\text{-}2 = \frac{\sum_{\Sigma \in \{ReferenceSummaries\}} \sum_{bi\text{-}gram \in \Sigma} count_{match}(bi\text{-}gram)}{sum_{\Sigma \in \{ReferenceSummaries\}} \sum_{bi\text{-}gram \in \Sigma} count(bi\text{-}gram)}$$

and the standard F1-measure. As both measures require manual summaries and having a single summary can be problematic [8], we used two sets of summaries (provided by domain experts, mimicking extractive summarization). For DUC2002, we selected [1] as a baseline since it obtained better results than SVM or CRF. [1] used a normalized Google distance (NGD) as similarity measure.

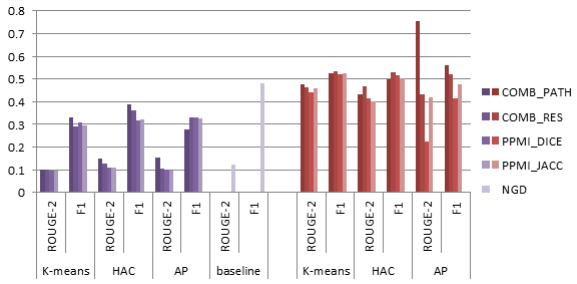

Fig. 2. Clustering results on the DUC2002 (violet) and HES–SO data sets (red)

Figure 2 shows the average results for three clustering techniques when both manual summaries are taken into account for the HES–SO and DUC2002 data sets. It can be seen that the same clustering methods perform differently on the two data sets, which is expected considering that they belong to different domains. Additionally, results vary both on similarity measures and clustering

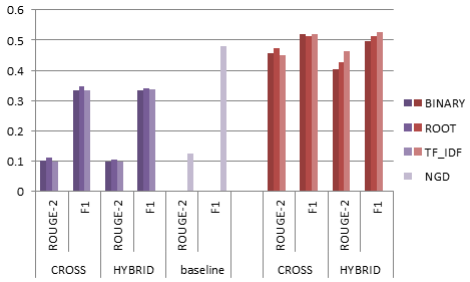

Fig. 3. LSA results on DUC2002 (violet) and HES–SO (red)

techniques. For DUC2002, better results were mainly obtained using a combined similarity measure with path similarity. For this similarity measure and two clustering methods: HAC and AP, we obtained better average ROUGE–2 results than the one provided by the baseline (0.15015 and 0.15155, versus 0.12368 respectively). At the same time, the F1 measure obtained (0.3893 for HAC and 0.2868 for AP) is worse than for the baseline (0.47947).

LSA results on both data sets (with both manual summaries) can be seen in Figure 3. The LSA CROSS method applied on the ROOT word–sentence matrix scored the best (average ROUGE-2 on DUC2002 was 0.11135). The best result on DUC2002 for the HYBRID method was also obtained for ROOT word–sentence matrix (0.10434). When two sets of manual summaries for the HES–SO data set are considered separately, results for the average ROUGE–2 vary (Figure 4). The smallest difference is achieved for LSA CROSS on the ROOT matrix, the highest for K–Means with a combined Resnik similarity. In general, results are better with the summary of the domain expert.

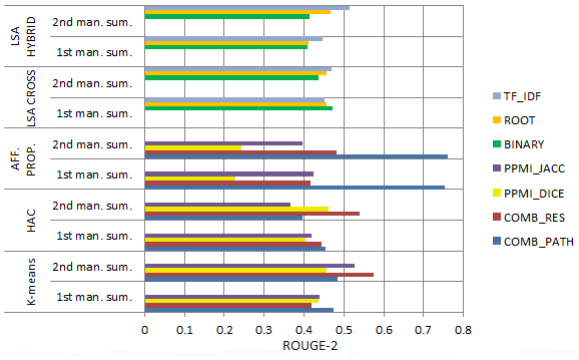

Fig. 4. Comparison of ROUGE–2 for different manual summaries (HES–SO data set)

5 Discussion and Conclusions

The reported reference extraction and parsing accuracy of 82%, better than 80% reported in [12], is suffient for further analysis but maybe not the maximum that is reachable. However, solving the mentioned exceptions was not the focus of this work. Using plain text resulted in difficulties to eliminate headers/footers, (sub)titles and even table/figure captions from the text, deteriorating citation extraction accuracy. Our ROUGE–2 results for HAC and AP with combined path similarity are higher than the compared baseline, which indicates that these methods generate summaries with correct bi–grams (as compared to manual summaries). On the other side, F1 (related to uni–gram matches and to ROUGE–1) is lower thn the baseline, so we can not be certain that the number of uni–grams is higher than the baseline. This situation may seem inconsistent but it actually indicates that our algorithms have the ability of generating summaries with a high number of overlapping bi–grams compared to manual summaries.

This work aims to help researchers reviewing scientific publications in a more efficient way by providing summaries of articles based on citation contexts. For this, we implement a novel workflow and carry out experiments applying several unsupervised extractive summarization techniques, based on clustering and LSA. We extend the claims of [1] and [16], demonstrating that not only similarity measures have impact on the summarization result but that different clustering techniques lead to different summarization results even when the same similarity measure is used. We show an improvement of the average ROUGE–2 measure on DUC2002 for HAC and AP clustering with a combined similarity measure using the WordNet path similarity. As future work, we consider using a medical thesaurus (e.g. MeSH) instead of general purpose WordNet.

References

1. Aliguliyev, R.: A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications 36(4), 7764–7772 (2009)
2. ark, D.: Automatic extraction of reference linking information from online documents. Tech. rep., Cornell University, Ithaca, NY, USA (2000)
3. Bird, S.: NLTK: The natural language toolkit. In: Proceedings of the Coling/ACL on Interactive Presentation Sessions. pp. 69–72. Stroudsburg, PA, USA (2006)
4. Bradshaw, S.: Reference directed indexing: Redeeming relevance for subject search in citation indexes. In: Research and Advanced Technology for Digital Libraries, pp. 499–510. Springer (2003)
5. Conroy, J., O'Leary, D.: Text summarization via hidden markov models. In: Proceedings of the 24th Annual International ACM SIGIR Conference. pp. 406–407. New York, NY, USA (2001)
6. Elkiss, A., Shen, S., Fader, A., Erkan, G., States, D., Radev, D.: Blind men and elephants: What do citation summaries tell us about a research article? Journal of the American Society Information Science and Technology 59(1), 51–62 (2008)
7. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science 315(5814), 972–976 (2007)

8. Jing, H., Barzilay, R., McKeown, K., Elhadad, M.: Summarization evaluation methods: Experiments and analysis. In: AAAI Symposium on Intelligent Summarization. pp. 51–59 (1998)
9. Jurafsky, D., Martin, J.: Speech & Language Processing. Pearson Education India (2000)
10. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. pp. 74–81 (2004)
11. Miller, G.: Wordnet: A lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
12. Haddou ou Moussa, K., Mayr, P.: Automatische referenzextraktion mit parscit. Social Media and Web Science - Das Web als Lebensraum, DGI pp. 425–428 (2012)
13. Nenkova, A., Passonneau, R.: Evaluating content selection in summarization: The pyramid method. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 145–152 (2004)
14. Niwa, Y., Nitta, Y.: Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In: Proceedings of the 15th Conference on Computational Linguistics. COLING '94, vol. 1, pp. 304–309 (1994)
15. Ozsoy, M.G., Cicekli, I., Alpaslan, F.N.: Text summarization of turkish texts using latent semantic analysis. In: Huang, C.R., Jurafsky, D. (eds.) Proceedings of the 23rd International Conference on Computational Linguistics. pp. 869–876. Tsinghua University Press (2010)
16. Pei-Ying, Z., Cun-He, L.: Automatic text summarization based on sentences clustering and extraction. In: Proceedings of 2nd IEEE International Conference on the Computer Science and Information Technology. pp. 167–170. IEEE (2009)
17. Qazvinian, V., Radev, D.: Scientific paper summarization using citation summary networks. In: Proceedings of the 22nd International Conference on Computational Linguistics. vol. 1, pp. 689–696 (2008)
18. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. vol. 1, pp. 448–453 (1995)
19. Ritchie, A., Robertson, S., Teufel, S.: Comparing citation contexts for information retrieval. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. pp. 213–222. CIKM '08, ACM, New York, NY, USA (2008)
20. Saad, S.M., Kamarudin, S.S.: Comparative analysis of similarity measures for sentence level semantic measurement of text. In: IEEE International Conference on Control System, Computing and Engineering. pp. 90–94. IEEE (2013)
21. Steinberger, J., Ježek, K.: Using latent semantic analysis in text summarization and summary evaluation. In: Proceedings of Industrial Management. pp. 93–100. ISIM '04 (2004)
22. Svore, K.M., Vanderwende, L., Burges, C.: Enhancing single-document summarization by combining ranknet and third-party sources. In: Proceedings of Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning. pp. 448–457 (2007)
23. Torres-Moreno, J.M., Saggion, H., da Cunha, I., SanJuan, E.: Summary Evaluation With and Without References. Polibits: Research journal on Computer science and computer engineering with applications 42, 13–19 (2010)
24. Zechner, K.: Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In: Proceedings of the 16th Conference on Computational Linguistics. vol. 2, pp. 986–989 (1996)