Analyzing Preprocessing Settings for Urdu Single-document Extractive Summarization

Muhammad Humayoun[1], **Hwanjo Yu**[2]
[1]IRIT (Institut de Recherche en Informatique de Toulouse), Université Paul Sabatier, Toulouse, France
[2]Department of Computer Science & Engineering, Pohang University of Science and Technology (POSTECH), South Korea
muhammad.humayoun@irit.fr, hwanjoyu@postech.ac.kr

Abstract

Preprocessing is a preliminary step in many fields including IR and NLP. The effect of basic preprocessing settings on English for text summarization is well-studied. However, there is no such effort found for the Urdu language (with the best of our knowledge). In this study, we analyze the effect of basic preprocessing settings for single-document text summarization for Urdu, on a benchmark corpus using various experiments. The analysis is performed using the *state-of-the-art* algorithms for extractive summarization and the effect of stopword removal, lemmatization, and stemming is analyzed. Results showed that these pre-processing settings improve the results.

Keywords: automatic text summarization, single-document summarization, extraction based summarization, benchmark experiments, preprocessing

1. Introduction

Urdu is an Indo-Aryan language, which is widely spoken[1]. Urdu script is an extended version of Perso-Arabic script. Similar to the other Perso-Arabic scripts, it is written right to left, in a complex, cursive-style writing systems. Urdu inherits a lot of vocabulary from Arabic, Persian and the native languages of South Asia (Humayoun et al., 2007). Due to this influence, Urdu has a complex morphology. In terms of syntax, it has a relatively free word order (Subject Object Verb). Despite spoken by millions of people, Urdu is an under-resourced language in terms of available computational resources.

1.1. Automatic Text Summarization

Generally, automatic text summarization addresses the issue of generating shortened information from a single document (or multiple documents written on the same topic). Usually, this shortened text is significantly less than the original text(s) but never more than half of the original text(s) in general (Radev et al., 2002). Two main approaches used for automatic text summarization are *abstraction based* and *extraction based*. Abstraction based approach extracts key points presented in different sentences and constructs a coherent summary from the original text by eliminating insignificant details. This requires solving hard questions such as semantic representation, inference, natural language generation, etc. (Radev et al., 2002). In contrast, extraction based technique for summarization is relatively straight forward. It relies on identifying most important sentences from the original document and assign them weights (based on their importance). Summary is then formed using top n sentences using these weights. The value of n depends on the length of required summary. These selected sentences are kept intact as units even when some of their parts may not contain most important information. For extraction based technique, several unsupervised methods have been suggested over the years. Some classic examples are (Luhn, 1958) and (Baxendale, 1958; Edmundson, 1969) that use word frequency and sentence position respectively. TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004) are among the famous and widely-used graph-based algorithms(Leite et al., 2007). We have used both of them in this work for the evaluation of pre-processing settings (Section 2.). To the best of our knowledge, there are only two studies for text summarization on Urdu (Burney et al., 2012; Patel et al., 2007). Both studies consider structural and statistical factors only. However, none of them analyze the effect of pre-processing settings.

Summarization algorithms are generally language-independent. However, customized pre-processing work, which is language dependent, is always required (Leite et al., 2007; Torres-Moreno, 2012a; Torres-Moreno, 2014) especially for morphologically rich languages (Nuzumlal and Özgür, 2014; Eryiğit et al., 2008). Examples of such pre-processing work could be identifying proper word boundary and sentence boundary. In addition, language-dependent resources such as stemmers, the lists of stopwords, lemmatizers, etc, may also be required to improve the quality of a generated summary. Therefore, analyzing pre-processing settings for Urdu, which is indeed morphologically rich, is an important research question that this paper tries to answer. These settings are discussed in Section 3..

1.2. Contribution

We have performed three benchmark experiments (Section 4.). First, we analyze the effect of four different stopword lists. Second, we analyze the effect of lemmatization (including rule-based and fixed-length stemming techniques). Third, we observe the effect of stopwords and lemmatization together. The experiments are run on *Urdu Summary Corpus* (Humayoun et al., 2016) which is a benchmark resource for single-document text summariza-

[1]Urdu has more than 100 million speakers, according to Ethnologue: `www.ethnologue.com/language/urd` Last visited: 06-03-2016

tion (Section 2.).
Urdu words cannot always be separated by spaces (Durrani and Hussain, 2010). Urdu Summary Corpus recognizes this problem and provides two versions of the same document collection; properly segmented and space segmented. This allowed us to measure the effect of proper word boundary identification for Urdu text summarization for all experiments (Section 4.1.1.). For evaluation, we used ROUGE (Lin, 2004) the *de facto* standard (Section 2.). The details of evaluation and results are given in Section 4.. Finally, Section 5. concludes the paper.

2. Background

2.1. Urdu Summary Corpus

Urdu Summary Corpus (Humayoun et al., 2016) is used in this study which is a benchmark corpus for single-document summarization[2]. It provides 50 articles and their corresponding human-written summaries (abstracts) covering various domains. More precisely, Urdu Summary Corpus consists of 1) fifty Urdu articles that were collected from various sources, and normalized, 2) fifty abstractive single-document summaries, (3) fifty part-of-speech tagged articles, 4) fifty morphologically analyzed articles, (5) fifty lemmatized articles, and (6) fifty stemmed articles. In addition, the basic NLP software tools such as, a normalizing utility, POS tagger, morphological analyzer, lemmatizer and stemmer are also provided.

2.2. Algorithms

The algorithms used in this study are (1) Degree Centrality (2) LexRank (3) Continuous LexRank (4) TextRank and (5) Baseline: Lead-based. The first three are proposed by (Erkan and Radev, 2004) and the fourth algorithm is proposed by (Mihalcea and Tarau, 2004). As a baseline, Lead-based method is used in which simply first n sentences are selected from the document as a summary.

2.3. Evaluation Methodology

For summary evaluation, ROUGE (Lin, 2004) has shown high correlation in content match with human evaluation (Liu and Liu, 2008). To calculate the similarity between two documents ROUGE provides five evaluation metrics. N-gram Co-Occurrence Statistics with Recall (ROUGE-N)(Lin and Hovy, 2003) and F1[3] measure of Longest Common Subsequence (ROUGE-L) (Lin and Och, 2004) are among them, and, we have used both of them for the evaluation in this study.

3. Pre-processing Settings

3.1. Lists of Stopwords

Stopwords are the frequent words in a language which only serve as syntactic function (Baeza-Yates and Rebiero-Neto, 2003). They are usually meaningless and belong to closed classes (Lo et al., 2005). We have prepared three lists of stopwords (in addition to an existing list) for the experiments.

- The first list is taken from (Burney et al., 2012). It contains 519 words.
- The second list is built by calculating the term frequency (Kenney and Keeping, 1962; Lo et al., 2005), on a large Urdu corpus (Jawaid et al., 2014). Nouns and Named entities are not included. It contains 500 words.
- Some studies stress the use of customized stopword lists that should be extracted from the domain corpus of the task in hand (Lo et al., 2005; Blanchard, 2007). Therefore, the third list is built by calculating the term frequency on the documents of Urdu summary Corpus (Humayoun et al., 2016). Nouns and Named entities are not included. It also contains 500 words.
- The fourth list contains only closed classes and has 195 words. The list is generated from the open source resources of Urdu morphology[4] (Humayoun et al., 2007).We built it using the open-source resources of Urdu morphology (Humayoun et al., 2007).

It is important to note that in all these stopword lists, words are separate on white-spaces.

3.2. Lemmatization and Stemming

We evaluated the effect of lemmatization and stemming (rule-based and fixed-length stemming). Urdu Summary Corpus provides lemmatized and stemmed articles produced by Urdu Morphological Analyzer (Humayoun et al., 2007) and Assas-Band stemmer (a rule-based Urdu Stemmer) (Akram et al., 2009) respectively. The coverage of lemmatization on Urdu Summary corpus is reported to be 64.5% on space segmented corpus and 65.2% in properly segmented corpus(Humayoun et al., 2016). For stemming, we do not find any statistics in Urdu Summary Corpus. However, Assas-Band seem to do considerable number of mistakes in stemming as suggested by the results in experiments 2 and 3 (Section 4.2. and 4.3. respectively).

3.2.1. Fixed-Length Stemming

It is a crude chopping, in which, the first n letters are kept and the rest are discarded. For instance, in the case of length $n = 1$ (FIX_1), we keep only the first letter[5]. FIX_1 is called ultra-stemming and it is shown that it may improve the quality of generated summaries(Torres-Moreno, 2012b). Concretely, we used the following two versions for the experiments:

1. FIX_n: Keeps first n letters. Words having the length less than n are kept intact. This may have over-stemming effect i.e. data sparseness problem (or noise).
2. SFIX_n: Keeps first n letters. Words having the length less than n are discarded. This way we ensure that some of the noise is discarded. Definitely, we may lose some valuable information as well.

[2]It is publicly available at `https://github.com/humsha/USCorpus`.

[3]The harmonic mean of precision and recall.

[4]Available at `http://www.lama.univ-savoie.fr/~humayoun/UrduMorph/`

[5]For example, in FIX_1 stemming, surface forms such as "run", "ran", . . . , and words like "rat", "race", . . . , are replaced by "r".

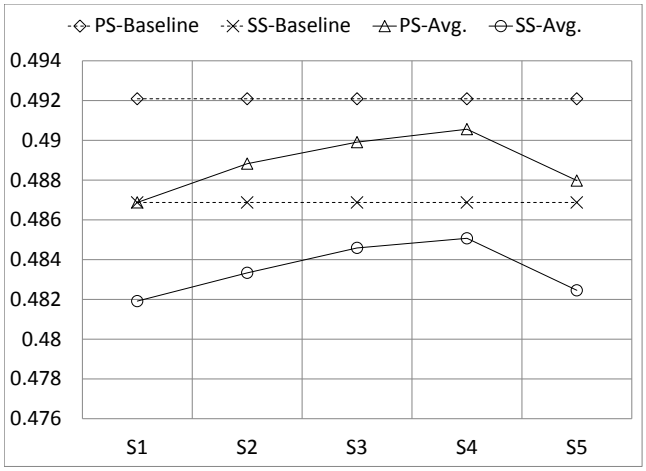


Figure 1: Experiment 1 – average results for both versions. PS: Properly Segmented, SS: Space Segmented. Correlation between them is 99%.
Sk on x-axis, stands for preprocessing setting k, e.g., S1 is the first preprocessing setting.
Similarity score is on y-axis. Though the variation in results is small but cannot be neglected; see 4..

4. Experimental Setup and Results

Every Human-Written (HW) summary is compared with the corresponding Machine-Written (MW) summary produced by an algorithm. Since we have five algorithms, we have five comparisons for each summary. This similarity comparison is performed by taking the average of ROUGE-N and ROUGE-L. For ROUGE-N, an average of unigram, bigram and trigram is used. Because of this setup, the evaluation metrics is quite strict in our opinion. Thus, even smaller variation in the results cannot be neglected. The similarity comparison is performed for all summaries and average is computed for the results in subsequent sections. Similarity scores are between 0 and 1.

4.1. Experiment 1: Effect of stopword Removal only

In this experiment, we measure the effect of stopwords removal on both versions of the Urdu summary corpus with the following preprocessing settings:

- S1: Raw – all tokens present, i.e., stopwords are not removed
- S2, S3, S4, S5: First, second, third, and fourth stopword list applied respectively

Scores for both versions of the corpus are computed for preprocessing settings S1 to S5. Figure 1, shows the average scores.
It is clear that the removal of stopwords improves results. We get best results for S4. This suggests that applying a domain specific stopword list gives best results (as suggested by many studies including (Lo et al., 2005; Blanchard, 2007)). However, a stopword list computed from a large balanced corpus also improves the results as shown in the preprocessing setting S3 – the second best.
Figure 2a shows the results for properly segmented corpus for each algorithm. TextRank outperforms in all preprocessing settings. However, no algorithm outperform the PS-baseline.
Figure 2b shows the results for space segmented corpus, having somewhat similar trends. For instance, TextRank is performing better for S3, S4 and S5. Again, no algorithm outperform the SS-baseline.

4.1.1. Effect of Word Segmentation

Effect of word segmentation in all experiments (including Experiment 2 and 3 mentioned in forthcoming sections) is given in Table 1. It suggests that proper word segmentation has the positive effect in experiment 1 with an average score 0.07. In contrast, proper word segmentation has a negligible effect for Experiment 2 and 3. From these results, it may be concluded that proper word segmentation improves the summarization results marginally.
However, it is worth note that the resources (stopwords lists, lemmatizer, and stemmer) are built on space segmented words. Therefore, it may be inferred that if the resources are built on properly-segmented words, we might have better results for PS summary corpus.

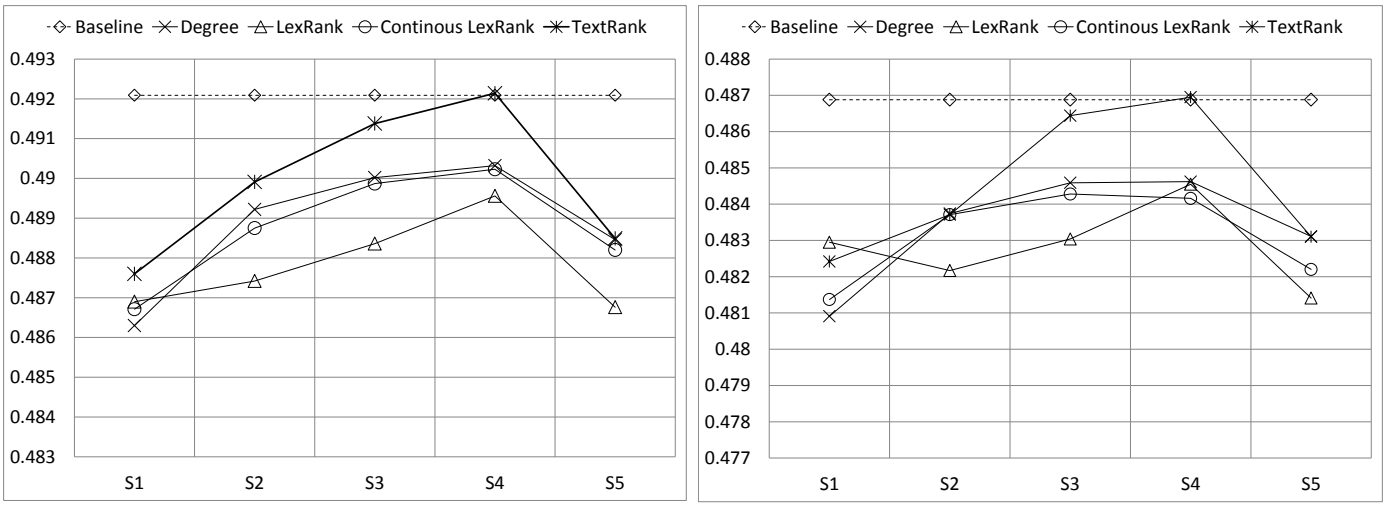
	E1-Avg	E2-Avg	E3-Avg	Average
Properly segmented	0.56	0.49	0.49	0.51
Space segmented	0.49	0.49	0.49	0.49
Difference	0.07	-0.0006	0.00004	0.023

Table 1: Word Segmentation Effect for all Experiments
E1:Experiment 1, E2:Experiment 2 and E3:Experiment 3

4.2. Experiment 2: Effect of Lemmatization and Stemming

In this experiment, we measure the effect of lemmatization and stemming on both versions of the summary corpus. Note that stopwords are not removed. Figure 3 shows the average results for both versions and compare them with the corresponding baselines. Pre-processing setting S1 (i.e. no preprocessing setting applied) is added for comparison. The figure suggests that the results for both versions correlate well with each other.
Humayoun's Lemmatizer (i.e. preprocessing setting S6) has positive effect if it is compared with the results when no preprocessing settings applied (i.e. S1). These results has outperformed the stemmer Assas-Band (i.e. S7) and different versions of the fixed length stemming (S9 to S17). It is worth noting that the Humayoun's lemmatizer has only 65% coverage on Urdu Summary Corpus. We argue that if the lemmatizer has more coverage, the results have been improved[6].
The results for the stemmer Assas-Band (i.e. S7) are worst for both versions of the corpus. This seem to suggest that the stemmer is doing many mistakes in stemming. This opinion is based on the observation that even applying no

[6]However, improvement with 100% coverage needs to be investigated in future.



(a) Properly Segmented Summary Corpus

(b) Space Segmented Summary Corpus

Figure 2: Results of Experiment 1

preprocessing (i.e. S1) is doing better than the Assas-Band stemmer.

The preprocessing setting S8 (i.e. fixed length stemming with length 1 – Fix_1) improves results, and, it scores best in Fix_1 to Fix_9 range. The trend of decreasing score in Fix_1 to Fix_9, suggests that we may have similar results for Fix_{10} similar to the Assas-Band stemmer (i.e. S7). The possible reason for the similar results may be that, probably, both have the same level of inconsistent over-stemming, causing data sparseness problem. In contrast, Fix_1 seem to have consistent over-stemming, which reduces sparseness.

The most prominent result is from the preprocessing settings S24 (i.e. $SFix_9$ stemming) for average. It outperforms the whole experimental settings including corresponding baselines for PS and SS corpus respectively.

When taking maximum value among algorithms (i.e. PS-Max and SS-Max in Figure 3), preprocessing setting S19 ($SFix_4$) and S20 ($SFix_5$) outperforms. It is worth noting that in Experiment 1, none of the pre-processing settings outperformed the baselines. This demonstrates that proper lemmatization or stemming may have more positive effect than stopword removal.

Figure 4a and Figure 4b show the results for PS and SS versions respectively. It is worth noting that the TextRank algorithm outperforms others for preprocessing setting S19 (i.e. $SFIX_4$), S20 (i.e. $SFIX_5$) and S24 (i.e. $SFIX_9$) for both versions. It indicates that for a resource-poor language like Urdu, summarization quality may be improved with fixed-length stemming (which is easy to implement).

4.3. Experiment 3: Combined Effect of Stopword removal and Lemmatization & Stemming

We measured the combined effect of stopword removal and stemming in this experiment. In Experiment 1, we observed that the pre-processing setting S3 has outperformed[7]. Thus, in this experiment, first, we remove the stopwords using this customized stopwords list (of the preprocessing setting S3), and then, apply the stemming approaches of Experiment 2. Figure 5 shows the average results for both versions and compare them with the corresponding baselines. The preprocessing settings S1 and S3 are also shown for a comparison. Average results for Experiment 2 are also displayed for an easy comparison. In experiment 2, the last preprocessing setting was S25. Thus, the preprocessing settings starts from S26 in this experiment.

For the combined effect of stopwords removal and applying Humayoun's Lemmatizer in preprocessing setting S26, the average results are slightly improved for both PS and SS versions of the corpus.

In the case of the combined effect of stopwords removal and Assas-Band stemming in preprocessing setting S27, the average results improve significantly as compared to the preprocessing setting S7 (i.e. when Assas-Band stemming was applied but stopwords were not removed) of Experiment 2. This may be affected by the fact that stopwords removal leaves fewer words, which as a result, reduces sparseness caused by the incorrect stemming. The same phenomena seem to happen for the preprocessing settings S29 to S34, i.e., results improve significantly.

Results also improve for the preprocessing setting S37 (i.e. $SFix_2$+stopword removal) and S38 (i.e. $SFix_3$+stopword removal). However, from preprocessing setting S39 (i.e. $SFix_4$+stopword removal) to S44 (i.e. $SFix_9$+stopword removal), we get mixed results; sometimes slight improvement and sometimes not. It is probably because SFix itself reduces the sparseness in model space and at some point it stops improving. On average, S44 (i.e. $SFIX_9$+stopword removal) and S24 (i.e. $SFIX_9$ of experiment 2) outperform

[7]Recall that, in this preprocessing setting, a customized stopwords list was applied to the document collection before producing machine generated summaries.

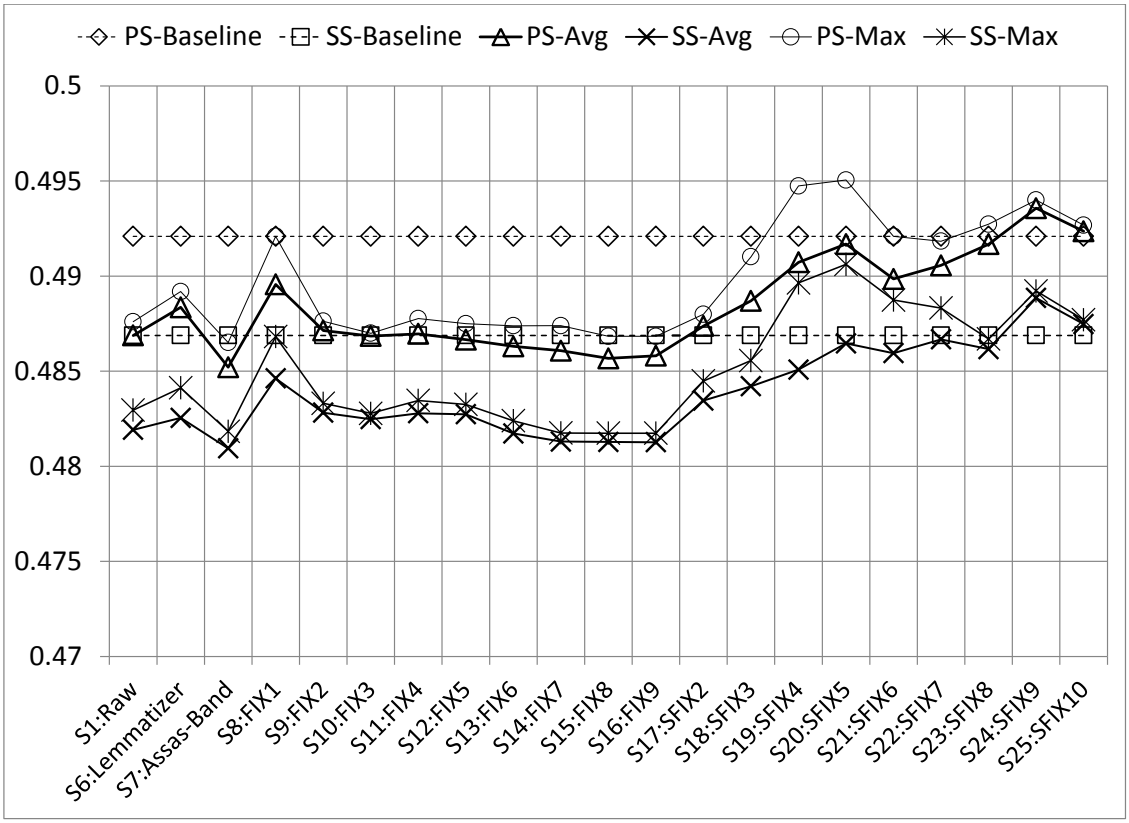
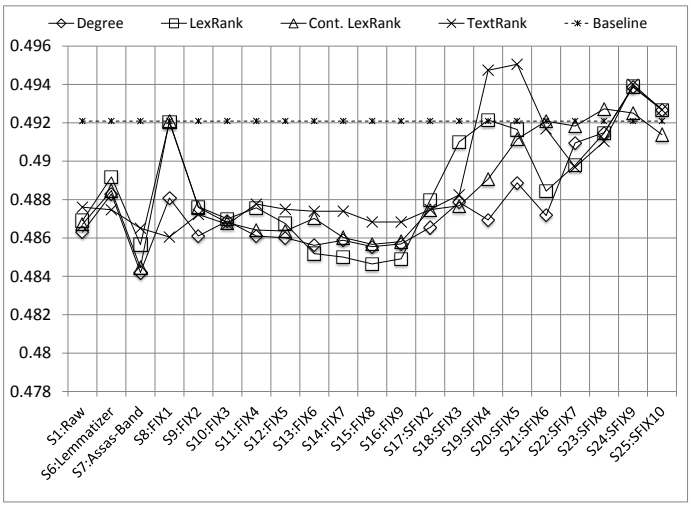
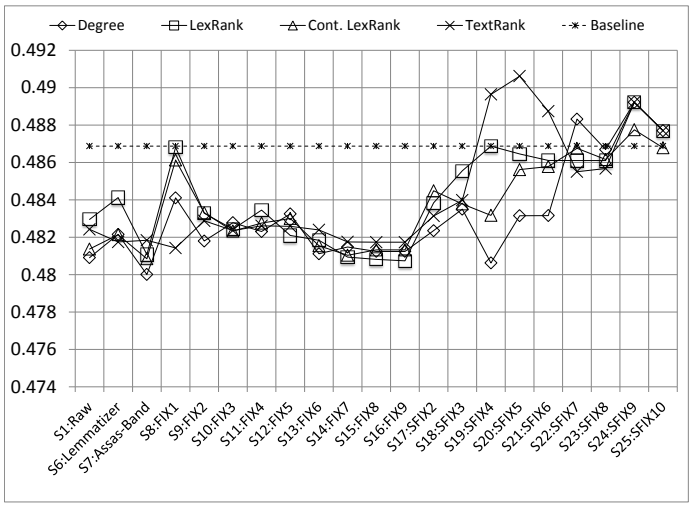


Figure 3: Experiment 2: average results for both versions. PS: Properly Segmented, SS: Space Segmented, Avg: average result of algorithms, Max: maximum value for a preprocessing setting among the algorithms



(a) Properly Segmented Summary Corpus



(b) Space Segmented Summary Corpus

Figure 4: Results of Experiment 2

all preprocessing settings (S1 to S45) equally.

Figure 6a and Figure 6b show the results for PS and SS versions respectively. The combined effect of stopwords removal and special fixed-length stemming with length 5 (i.e. SFix5) in preprocessing setting S40 outperforms all other settings for TexRank algorithm.

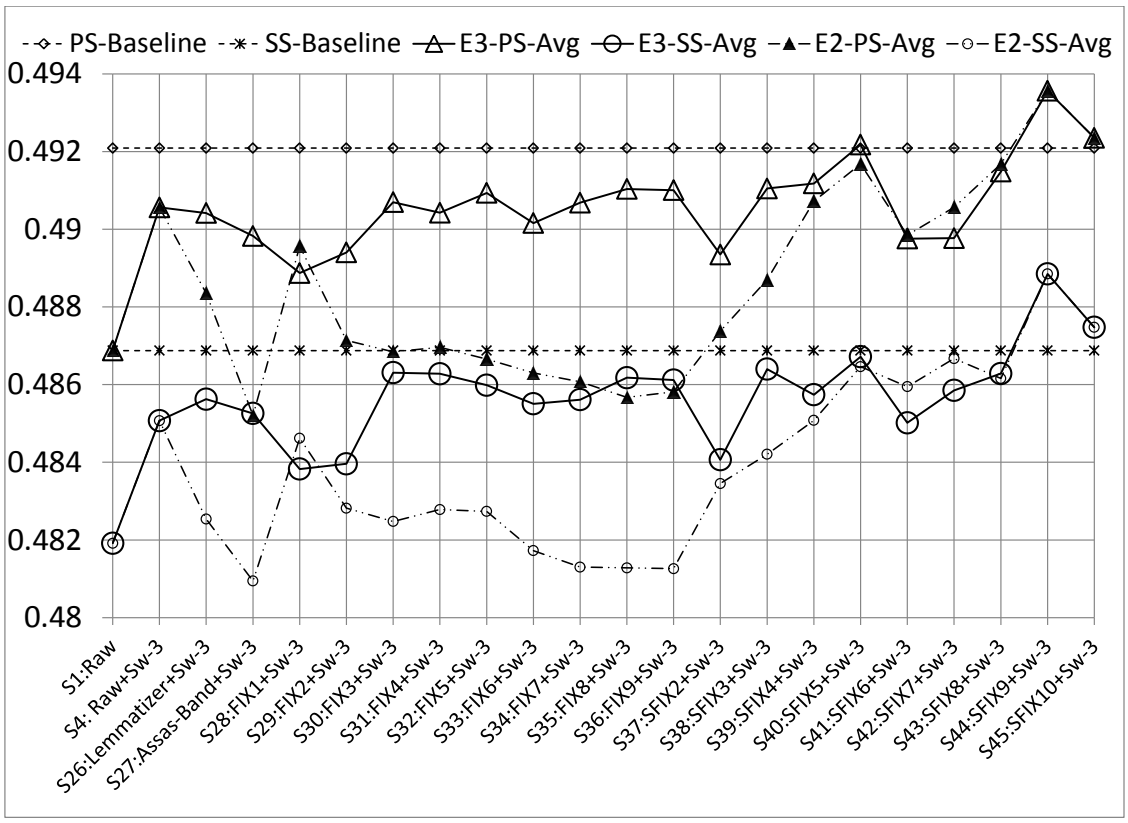
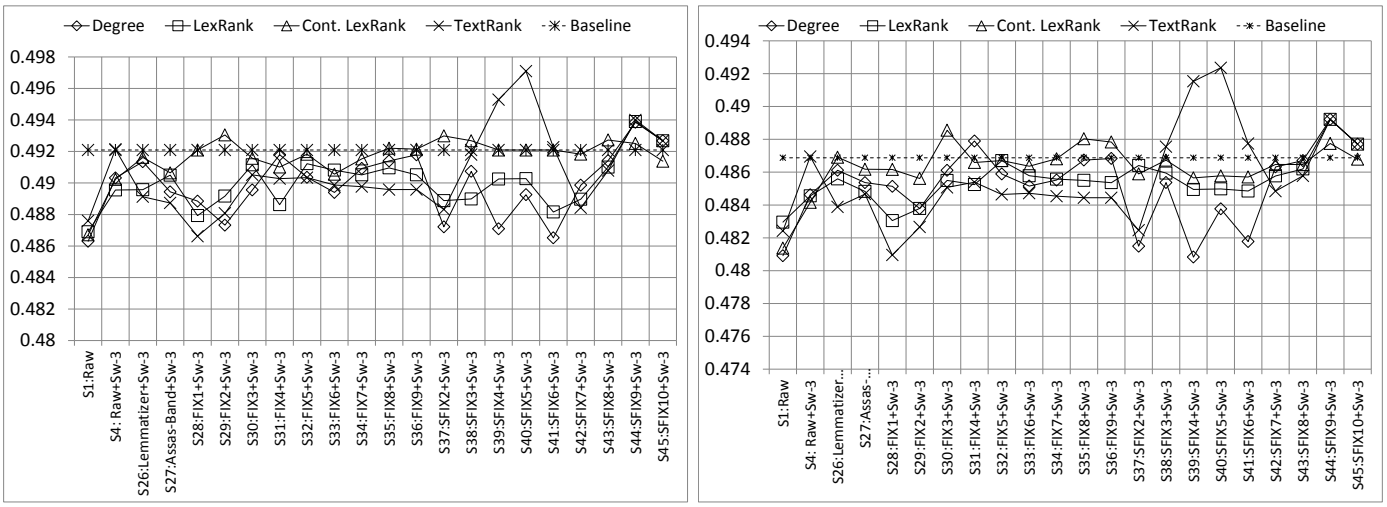


Figure 5: Experiment 3: Average Results for both versions compared with Experiment 2. E2: Experiment 2, E3: Experiment 3, Sw-3: customized stopwords list from preprocessing setting S3, S26:Lemmatizer+Sw-3: preprocessing setting number 26 in which stopwords are removed in addition to applying lemmatization



(a) Properly Segmented Summary Corpus

(b) Space Segmented Summary Corpus

Figure 6: Results of Experiment 3

5. Conclusion

This paper reports an analysis of various pre-processing settings for single-document text summarization for the Urdu language on a freely available benchmark summary corpus (Humayoun et al., 2016). We used state-of-the-art algorithms in these experiments. It seems that the results are marginally different from each other. However, we take the average of two evaluation measures ROUGE-N[8] and ROUGE-L, and thus, even smaller variation in results cannot be neglected (in our opinion).

The results suggest that applying proper word segmentation improves the results if stemming is not applied. However, if stemming is already planned (with or without stopwords removal) then words can be tokenized on spaces to get the same results. This seems promising in the sense that currently applying proper word segmentation is not trivial due to the lack of readily available word segmentation software for Urdu. We observed that applying stopword lists alone improve results on both versions of the corpus. Among three stopwords lists, the maximum improvement is achieved when a customized list is used.

For lemmatization and stemming, we observed the limitations of existing resources. It is observed that incorrect stemming done by the Assas-Band stemmer and limited coverage of the Humayoun's lemmatizer undermines the results. Thus, these resources must be improved. In contrast, some versions of the fixed-length stemming, such as $SFix_9$, has performed best for all the algorithms on average. Fixed-length stemming is easy to implement. So it is a positive indicator for the under-resourced language such as Urdu. We observed that the effect of stemming is more significant than the stopwords removal. Finally, stemming and stopword removal together improves the results even further for many stemming approaches.

6. Bibliographical References

Akram, Q.-u.-A., Naseer, A., and Hussain, S., (2009). *Proceedings of the 7th Workshop on Asian Language Resources (ALR7)*, chapter Assas-band, an Affix-Exception-List Based Urdu Stemmer, pages 40–47. Association for Computational Linguistics.

Baeza-Yates, R. and Rebiero-Neto, B. (2003). *Modern Information Retrieval*. Addison Wesley, London, England.

Baxendale, P. B. (1958). Machine-made index for technical literature: An experiment. *IBM Journal of Research and Development*, 2(4):354–361, October.

Blanchard, A. (2007). Understanding and customizing stopword lists for enhanced patent mapping. *World Patent Information*, 29(4):308–316.

Burney, A., Sami, B., Mahmood, N., Abbas, Z., and Rizwan, K. (2012). Urdu text summarizer using sentence weight algorithm for word processors. *International Journal of Computer Applications*, 46(19), May. ISSN: 0975 - 8887.

Durrani, N. and Hussain, S. (2010). Urdu word segmentation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 528–536.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of ACM*, 16(2):264–285, April.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial intelligence research (JAIR)*, 22(1):457–479.

Eryiğit, G., Nivre, J., and Oflazer, K. (2008). Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389.

Humayoun, M., Hammarström, H., and Ranta, A. (2007). Urdu morphology, orthography and lexicon extraction. *CAASL-2: The Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA Linguistic Institute. Stanford University, California, USA.*, pages 21–22. `http://www.lama.univ-savoie.fr/~humayoun/UrduMorph/`.

Humayoun, M., Nawab, R. M. A., Uzair, M., Aslam, S., and Farzand, O. (2016). Urdu summary corpus. Submitted at LREC 2016. Publicly available at `https://github.com/humsha/USCorpus`.

Jawaid, B., Kamran, A., and Bojar, O. (2014). A tagged corpus and a tagger for urdu. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Kenney, J. F. and Keeping, E. S. (1962). *Mathematics of Statistics, Part 1*. Princeton, NJ: Van Nostrand Reinhold., 3rd edition.

Leite, D. S., Rino, L. H. M., Pardo, T. A. S., Gracas, M., and Nunes, V. (2007). Extractive automatic summarization: Does more linguistic knowledge make a difference? In C. Biemann, et al., editors, *Proceedings of the HLT/NAACL Workshop on TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, pages 17–24, Rochester, NY, USA. ACL.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 605–612, Barcelona, Spain, July.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Liu, F. and Liu, Y. (2008). Correlation between rouge and

[8] For ROUGE-N, an average of unigram, bigram and trigram is used.

human evaluation of extractive meeting summaries. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 201–204, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lo, R. T., He, B., and Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. *Journal of Digital Information Management (JDIM)*, 3(1):3–8.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, April.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In Dekang Lin et al., editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Nuzumlal, M. Y. and Özgür, A. (2014). Analyzing stemming approaches for turkish multi-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 702–706, Doha, Qatar, October. Association for Computational Linguistics.

Patel, A., Siddiqui, T., and Tiwary, U. S. (2007). A language independent approach to multilingual text summarization. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 123–132, Paris, France.

Radev, D. R., Hovy, E., and McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics*, 28(4):399–408, December.

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520.

Torres-Moreno, J. (2012a). Artex is another text summarizer. *CoRR*, abs/1210.3312.

Torres-Moreno, J. (2012b). Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. *ACM Computing Research Repository (CoRR)*, abs/1209.3126.

Torres-Moreno, J. (2014). Three statistical summarizers at CLEF-INEX 2013 Tweet contextualization track. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 565–573.