DegExt – A Language-Independent Keyphrase Extractor

Marina Litvak[1], Mark Last[2], and Abraham Kandel[3]

[1] Department of Software Engineering,
Sami Shamoon Academic College of Engineering
Beer-Sheva 84100, Israel
[2] Department of Information System Engineering,
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
[3] Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620, USA

Abstract. In this paper, we introduce DegExt, a graph-based language-independent keyphrase extractor,which extends the keyword extraction method described in (Litvak & Last, 2008). We compare DegExt with two state-of-the-art approaches to keyphrase extraction: GenEx (Turney, 2000) and TextRank (Mihalcea & Tarau, 2004). We evaluated DegExt on collections of benchmark summaries in two different languages: English and Hebrew. Our experiments on the English corpus show that DegExt significantly outperforms TextRank and GenEx in terms of precision and area under curve (AUC) for summaries of 15 keyphrases or more at the expense of a mostly non-significant decrease in recall and F-measure, when the extracted phrases are matched against gold standard collection. Due to DegExt's tendency to extract bigger phrases than GenEx and TextRank, when the single extracted words are considered, DegExt outperforms them both in terms of recall and F-measure. In the Hebrew corpus, DegExt performs the same as TextRank disregarding the number of keyphrases. An additional experiment shows that DegExt applied to the TextRank representation graphs outperforms the other systems in the text classification task. For documents in both languages, DegExt surpasses both GenEx and TextRank in terms of implementation simplicity and computational complexity.

Keywords: Keyphrase extraction, summarization, text mining, graph-based document representation

1 Introduction

Keyphrase extraction is defined as the automatic identification of a set of terms that can be used to describe a given document. The extracted keyphrases can be used to build an automatic index for a document collection, and they can also be used for document representation in categorization or classification tasks. In

addition, the set of keyphrases extracted from a certain document can function as an extractive summary of that document.

In this paper, we present a graph-based extractor DegExt and compare it to two other approaches for keyphrase extraction—TextRank and GenEx—that are used in the extractive summarization of text documents. According to our problem statement, viable keyphrases are those that are included in a gold standard document summary set created by human experts (e.g., DUC 2002).

In the 1950s, Luhn (1958) introduced a simple approach, based on using a frequency criterion, for selecting a document's keywords. Today, the state-of-the-art in keyword selection is represented by supervised learning methods, according to which a system is trained, based on lexical and syntactic features, to recognize keywords in a text.

The supervised learning approach for keyphrase extraction, first suggested by Turney (2000), entails the combination of parameterized heuristic rules with a genetic algorithm (GA) to create the GenEx system, which automatically identifies keywords in a document. GenEx uses a GA to learn the best parameters for the extractor algorithm, with parameter values for the extractor as the population and the precision of the extractor as the fitness function. GenEx is based on the traditional vector-space model and is language-dependent: as a supervised algorithm, it should be adapted to new languages or domains by retraining on every new type of data, which requires a high-quality corpus of annotated documents. Fortunately, Turney has shown that GenEx does generalize well to other domains.

Witten et al. (1999) introduced Kea, another supervised approach using a Naïve Bayesian Decision rule with two features: tf-idf and the distance of the word from the beginning of the text.

Hulth (2003) improved keyword extraction with a machine learning algorithm by adding linguistic knowledge (such as syntactic features) to the document representation instead of relying exclusively on statistics. The author showed that the results of any selection approach can be dramatically improved by extracting NP-chunks instead of n-grams and by adding the POS tag(s) assigned to each term as a feature.

All of the above approaches are supervised, and each uses a classic vector-space model for document representation. In contrast, Mihalcea and Tarau (2004) introduced an unsupervised, graph-based keyphrase extractor called TextRank. TextRank utilizes a simple, syntactic graph-based representation for text documents, where nodes stand for unique non-stop words (more precisely, lexical units of a certain part of speech) connected by undirected edges representing a *co-occurrence* relation, controlled by the distance between word occurrences: two vertices are directly connected if their corresponding words co-occur within a window of maximum N words, where $N \in [2, 10]$[4]. The main advantage of syntactic representation is its language-independence (given no syntactic filters) and simplicity—syntactic representation requires almost no language-specific linguistic processing. Mihalcea and Tarau (2004) remark that vertices added to the

[4] Best results shown for $N = 2$.

graph can be restricted with syntactic filters, which select only lexical units of a certain part of speech. But for multilingual processing, TextRank can be run without syntactic filtering during the formative stages of document representation, and therefore, all words can be considered. Since we designed our experiments based on the results of Mihalcea and Tarau (2004), we ran TextRank with a syntactic filter that focused only on nouns and adjectives as the best filter according to their results. Vertex importance within a graph was determined using PageRank, a graph-based ranking algorithm (Brin & Page, 1998). Formally, given the document graph $G(V, E)$, the score of a vertex V_i is defined by the recursive formula:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|}$$

where $In(V_i)$ is the set of vertices that connect with vertex V_i, $Out(V_i)$ is the set of vertices that vertex V_i connects with (successors), and d is a damping factor that integrates, into the model, the probability of jumping from a given vertex to another random vertex in the graph. Usually, the damping factor is set to 0.85, which is the value used in TextRank implementation. The top ranked vertices are extracted as the keywords. Post-processing, where adjacent document keywords are collapsed into multi-word keyphrases, is then performed.

Given the claim that for an undirected graph, PageRank is equivalent to degree centrality[5] (see the definition of degree centrality in footnote 7), we showed in (Litvak & Last, 2008) that applying ranking algorithms to document graphs (even directed ones) does not improve the extractor performance, and even makes it worse. Our latter work (Litvak et al., 2011) extends the approach based on degree centrality to the keyphrase extraction.

Recent papers explore World Wide Web knowledge like Wikipedia in order to model documents as semantic networks. Grineva et al. (2009) introduce title-community approach that uses the Girvan-Newman algorithm to cluster phrases into communities and selects those phrases in the communities containing the title phrases as key phrases. Li et al. (2010) propose a novel semi-supervised keyphrase extraction approach by computing the phrase importance in the semantic network, through which the influence of title phrases is propagated to the other phrases iteratively.

This paper extends our published conference paper (Litvak et al., 2011) in the following ways:

1. The description of the related work is enhanced.
2. Another language, Hebrew, is added to experiments.
3. New experiments and visual representations are added, according to reviews.
4. The experiments section, conclusions and future work are extended accordingly.

[5] Based on this fact, every application of PageRank to undirected unweighted graphs may be replaced by much lighter approach based on the degree centrality ranking.

2 DegExt — Degree-based Extractor

Like TextRank, DegExt is an unsupervised, graph-based, language-independent keyphrase extractor. DegExt uses graph representation based on the so-called *simple* graph-based syntactic representation of text and web documents defined in (Schenker et al., 2005), which enhances the traditional vector-space model by taking into account some structural document features. Under the *simple* graph representation words are represented by labeled nodes and unlabeled edges represent order-relationship between the words. The stemming and stopword removal operations of basic text preprocessing are performed before graph building[6] to reduce the number of graph nodes. A single vertex is created for each distinct word, even if the word appears more than once in the text. Thus, each vertex label in the graph is unique.

Unlike the original *simple* representation, where only a specified number of most frequent terms are added into graph, we don't have any restrictions on the number of graph nodes. In our system, filtering of nodes may be specified by configurable parameters using the absolute number of nodes or ratio threshold. However, in our experiments we did not limit the number of nodes at all, in order to avoid the dependency on additional parameter.

Edges represent order-relationships between two terms: there is a directed edge from A to B if an A's term immediately precedes a B's term in any sentence of the document. However, in the event that sentence-terminating punctuation marks (periods, question marks, and exclamation points) are present between the two words, an edge is not created. In order to adapt the graph representation to multi-word keyphrase extraction, we label each edge by the IDs of sentences that contain both words in the specified order. This definition of graph edges is slightly different from co-occurrence relations used in (Mihalcea & Tarau, 2004) for building undirected document graphs holding unlabeled edges, where the order of word occurrence is ignored and the size of the co-occurrence window varies between 2 and 10. Shortly, the "simple" representation is equivalent to TextRank with $N = 2$ and directed edges.

Figure 1 shows a sample text (three numbered sentences) and its graph representation, respectively.

The syntactic graph-based representations were shown by Schenker et al. (2004) to perform better than the classical vector-space model on several clustering and classification tasks. We chose to use the *simple* representation as a basis in this work because it is relatively cheap in terms of processing time and memory resources while having provided nearly the best results for the two above text mining tasks.

The nodes with the high *degree centrality*[7] in a document graph are assumed by DegExt to represent the keywords. When document representation is com-

[6] This part may be skipped for multilingual processing unless appropriate stemmers and stopword lists are provided for different languages.

[7] Degree centrality assigns importance to a node proportional to its degree – the number of edges incident to it. In a directed graph we can make a distinction between the in-degree (the number input arcs) and the out-degree (number of output arcs).

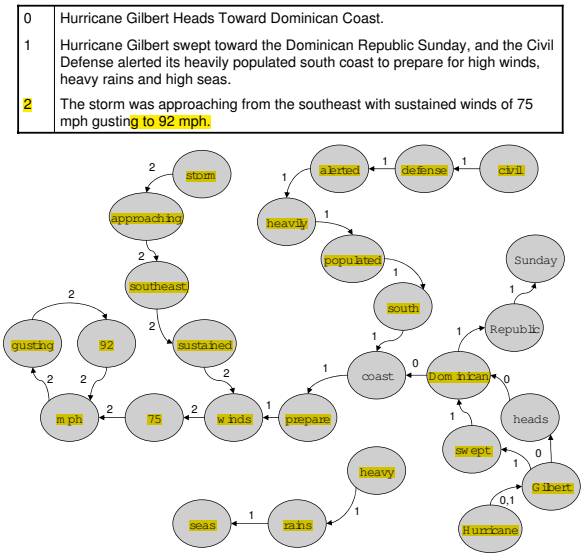

Fig. 1. Text and its graph representation.

plete, every node is ranked by the extent of its degree, and the top ranked nodes are then extracted. Intuitively, the most connected nodes, i.e., the top ranked nodes, represent the most salient words. According to the above representation, words that appear in many sentences that diverge contextually (i.e., the surrounding words change from sentence to sentence) will be represented by nodes with high degree centrality. This intuition was approved by experimental results in (Litvak & Last, 2008), where different number of HITS iterations was tried and the best result was achieved after the first iteration, ranking nodes by their degree. Thus, we showed that applying ranking algorithms to document graphs (using the *simple*-based representation) does not improve the extractor performance on English corpora, but even makes it worse.

In order to identify keyphrases (as sequences of adjacent keywords), DegExt scans the document graph during postprocessing marking all selected potential keywords in the graph, and sequences of adjacent keywords (up to 3) having the same label (sentence ID) on edges between them are combined into multi-word keyphrases. The postprocessing proposed by Mihalcea and Tarau (2004) is also applicable. The final rank for each phrase is calculated as an average between the ranks for each of its words. N (specified by the user) top-ranked phrases are extracted as keyphrases. Algorithm 1 demonstrates the DegExt pseudocode. Set of sequences of adjacent nodes connected by edges with the same label (P) can be computed by BFS($\check{G}, v$) for each $v \in \check{V}$ limited to 3 levels[8].

In this case, the degree of the node consists in the sum of the in-degree and of the out-degree of this node.

[8] Since we limit our phrases to 3 words at most, maximal length of a keyphrase cannot exceed 3.

Algorithm 1 DegExt

INPUT: text document D, maximal number of keyphrases N, degree threshold T
OUTPUT: list of keyphrases for D
$G(V, E) \Leftarrow$ graph representation for D
$\breve{G}(\breve{V}, \breve{E}) \Leftarrow$ graph representation for D after removal all nodes with degree $< T$
initialize phrase list $L \Leftarrow \emptyset$
$P \Leftarrow$ sequences of adjacent nodes connected by edges with the same label
for all $p \in P$ **do**
  $rank(p) =$ average degree of its nodes
  $L \Leftarrow L \bigcup p$
end for
sort L by rank in descending order
return top N phrases

Since the DegExt algorithm involves constructing document representation, sorting graph nodes by degree, and identifying top-ranked keyphrases, it has much lower computational complexity than TextRank, which needs additional time $O(c(|E|+|V|))$ to run the PageRank algorithm. Here c is the number of iterations needed to converge, $|E|$ is the number of edges, and $|V|$ is the number of nodes (words) in a document graph. Representation building in both algorithms has the same computational complexity (linear in the document length, assuming efficient implementation). When DegExt is applied to graph-based document representations constructed without syntactic filtering, it is truly language-independent.

3 Experiments

3.1 Experimental Setup

All experiments were performed on benchmark collections of summarized news articles in two languages: English and Hebrew. We intentionally chose these languages, which belong to distinct language families (Indo-European and Semitic languages, respectively), to examine the generalizability of our evaluation results.

For the English language, we used a corpus provided by the 2002 Document Understanding Conference (DUC) (DUC, 2002). This collection contains 533 English texts, each with an average of 2-3 abstracts (gold standard abstracts) per document.

For the Hebrew language, we generated a corpus of 50 news articles of 250 to 830 words each from the Website of the *Haaretz* newspaper[9] that were summarized by human assessors. After the quality assessment, comprised of comparing each assessor's summary to those of all the other assessors and checking the time spent on summarization, the final corpus of summarized Hebrew texts was compiled from the summaries of about 60% of the most consistent assessors (from 70 participants, in total), with an average of seven extracts of approximately 100

[9] `http://www.haaretz.co.il`

words, per single document[10]. The details about the experiment and the corpus can be found in (Litvak et al., 2010a; Litvak et al., 2010b), where it was used for evaluating the sentence extraction task.

Our evaluations aim to measure the quality of extracted keyphrases by performing the following:

- To evaluate the extraction results, we ran the keyphrase extractors on the document collection and compared the extracted keyphrases against the gold standard summaries (abstracts or extracts). We used common metrics such as *precision*, *recall*, *F-measure* and *AUC (area under curve)*. Extracted keyphrases that appeared in at least one abstract for a given document were considered *true positives*, extracted keyphrases that did not appear in any abstract were *false positives*, keyphrases that were not extracted and that did appear in the abstracts were considered *false negatives*, and keyphrases that were not extracted and that did not appear in abstracts were considered *true negatives*.
- We compared DegExt with the state-of-the-art approaches: GenEx and TextRank. Statistical test based on the normal approximation of the binomial distribution was performed to estimate the difference between DegExt and other systems performance.
- We evaluated the multilingual performance of DegExt by applying it on two languages: English and Hebrew. We compared its multilingual performance with TextRank as the state-of-the-art language-independent approach.
- We evaluated the appropriateness of the extracted keyphrases for the text classification task, using five different classifiers and comparing between four approach: DegExt, TextRank, GenEx and the hybrid approach integrating the DegExt with a TextRank representation (denoted by DegExt-TR).

3.2 Preprocessing and Implementation Details

In addition to stemming, we performed POS tagging of input texts (in order to be consistent with the competing system - TextRank) using the Stanford Log-linear Part-Of-Speech Tagger,[11] and a POS tagger embedded in Hebrew text analysis tool developed by BGU NLP group[12] (Adler, 2007; Goldberg et al., 2009), for English and Hebrew documents, respectively. Stopwords removal was performed only for the English corpus.[13]

After all preprocessing the xml-formatted files representing graphs were generated for each document in both corpora, and served as input for our keyphrase extractor. The size of syntactic graphs[14] extracted from English texts is 212 nodes on average, varying from 66 to 944, and for Hebrew texts is 323 nodes on average, varying from 193 to 562.

[10] Dataset is available at `http://www.cs.bgu.ac.il/~litvakm/research/`
[11] `http://nlp.stanford.edu/software/tagger.shtml`
[12] `http://www.cs.bgu.ac.il/~adlerm/freespace/tagger.zip`
[13] No standard stopword list for Hebrew exists.
[14] We define the size of a graph as the number of its vertices.

Since all evaluated extractors output multi-word phrases along with single words, we created an inverted index of phrases from the gold standard abstracts for comparison purposes.

3.3 Experimental Results

We compared DegExt with two state-of-the-art approaches for keyphrase extraction: GenEx (Turney, 2000) and TextRank (Mihalcea & Tarau, 2004). In order to evaluate the extraction results, we ran all three extractors on DUC 2002 and two of them, DegExt and TextRank,[15] on the Hebrew document collection and compared the extracted keyphrases against their gold standard abstracts (DUC 2002) or extracts (the Hebrew collection).

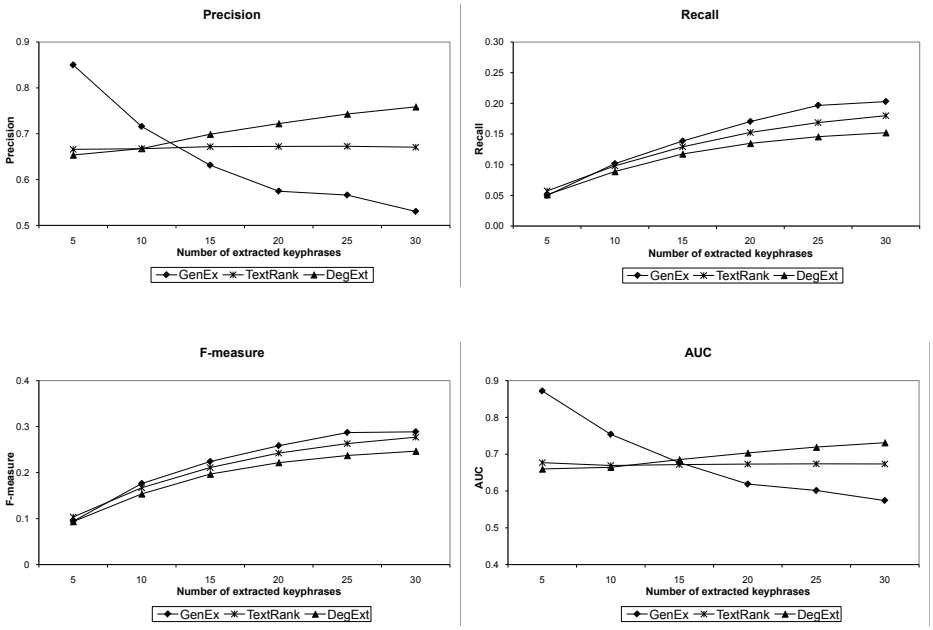

Fig. 2. English: evaluation results for GenEx, TextRank, DegExt and six models respectively (5 - 30 keyphrases).

Figure 3 presents five resulting keyphrases for TextRank, GenEx, and DegExt, respectively, in one of the English documents entitled "Statistics on Six Decades of Oscar With PM-Oscar Nominations" (in boldface). Figure 2 demonstrates the

[15] Since the GenEx is commercial and language-specific tool (we have English version), we were unable to adapt its API to Hebrew, whereas DegExt and TextRank implementations were adapted to both languages.

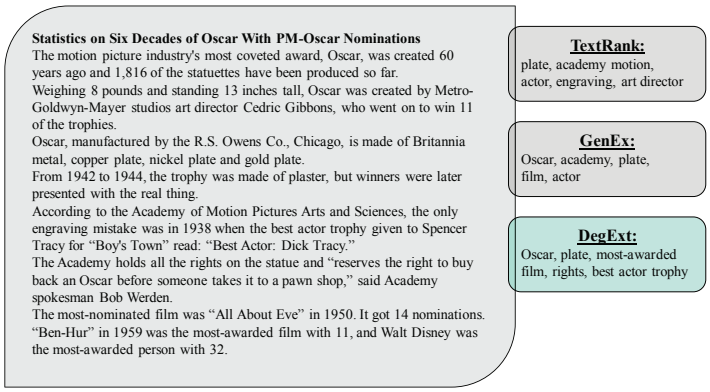

Fig. 3. English: sample extraction results for $N = 5$ and text document from DUC 2002 collection titled "Statistics on Six Decades of Oscar With PM-Oscar Nominations".

average[16] precision, recall, F-measure, and AUC values for each of the methods evaluated on the entire English corpus. We used macro-averaging in our calculations. For all methods, we considered six summary models distinguished by the number of top ranked phrases N—from 5 to 30, in steps of 5.

GenEx had the highest precision and AUC values for "small" models (up to 10 and 15 keyphrases, respectively). The higher precision value for GenEx can be explained by using the precision as a fitness function in the GA. Also, GenEx had the highest, but not statistically distinguishable, recall and F-measure results. Since GenEx does not always succeed to extract as many keypharses as required, its precision decreases with the number of needed keyphrases.

DegExt exhibited the best values for precision and AUC for the "large" models that extract greater numbers of required keyphrases (above 10 and 15 keyphrases, respectively), but those high values were obtained at the expense of relatively low recall and F-measure values. For example, for 20 extracted keyphrases, the F-measure of DegExt was approximately 15% and 10% lower than the highest (GenEx) and the second highest (TextRank) values, respectively. However, DegExt precision was approximately 30% and 15% better than the lowest (GenEx) and the second lowest (TextRank) values, respectively.

According to the normal approximation of the binomial distribution with $\alpha = 0.05$, applied to precision, DegExt significantly outperforms both extractors for a large number of extracted keyphrases (at least 25 and 15 for TextRank and GenEx, respectively). GenEx significantly surpasses DegExt in terms of precision only for one model – 5 extracted keyphrases, and in terms of recall for 25 and more extracted keyphrases. Both TextRank and DegExt have undistinguishable performance in terms of recall.

Figure 5 demonstrates the evaluation results on the Hebrew corpus. We can see that DegExt and TextRank perform differently in Hebrew: TextRank has the highest values for precision and AUC for all models at the expense of lower

[16] We used *macro-averaging* in our calculations.

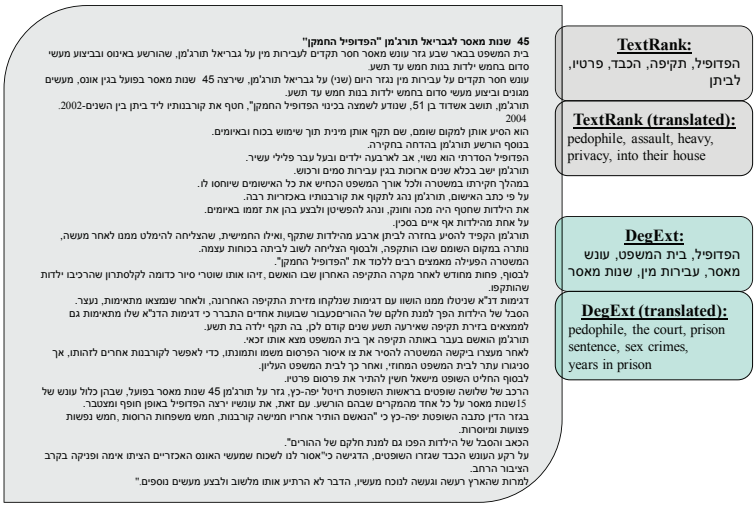

Fig. 4. Hebrew: extraction results for $N = 5$ and a text document titled "45 years in prison for Gabriel Turgeman, "elusive pedophile"" (translation).

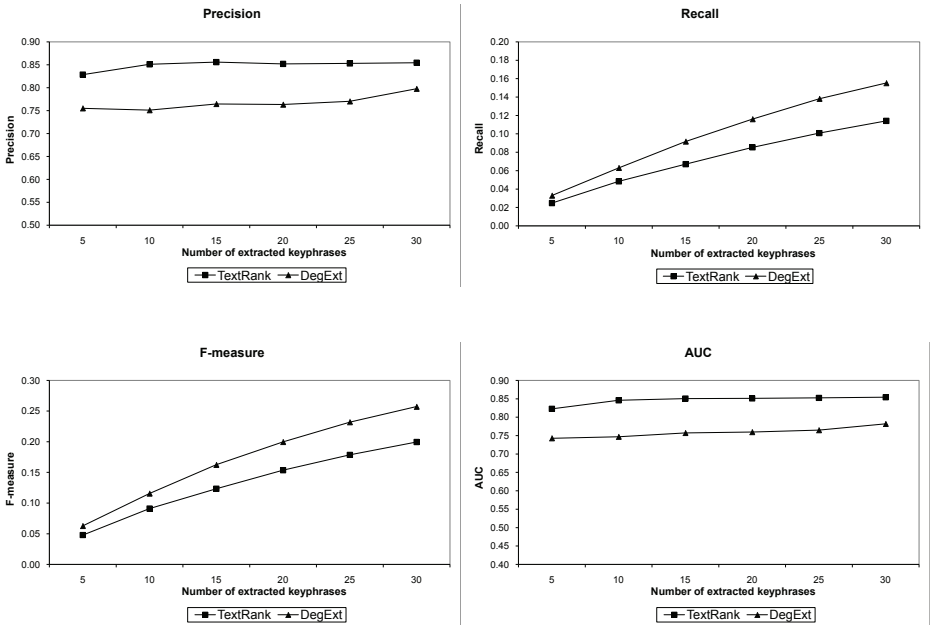

Fig. 5. Hebrew: evaluation results for TextRank and DegExt and six models respectively (5 - 30 keyphrases).

values of recall and F-measure. For example, given 20 as the number of extracted keyphrases, DegExt has a lower precision approximately by 10% than the TextRank value, and a better F-measure by approximately 25% than the TextRank value. According to the normal approximation of the binomial distribution with $\alpha = 0.05$, applied on both precision and recall, both methods perform the same disregarding the number of keyphrases. We explain the differences between English and Hebrew results by the following possible reasons:

(1) Morphological differences between the two languages. Different levels of morphological disambiguation in two languages (Adler, 2007) cause the difficulties in the matching between tagsets. Since there is no exact matching rules between POS tags in Hebrew and English, our rules for mapping *noun, adjective* and *verb* tags (we tried to be consistent with TextRank by using the same tagset) into Hebrew tagset may introduce some noise into the results. The same differences affect word features used by the extraction model.

(2) Differences in the preprocessing for two languages. For example, we did not remove stopwords from the Hebrew documents, that are frequently prefixes and suffixes of non-stop words. Consequently, the same unique word in Hebrew could be represented by several different graph nodes.

(3) Different performance of POS taggers for two languages: 97.24% of token accuracy for Stanford tagger on the English test set (Toutanova et al., 2003), and 93% of accuracy for the Hebrew POS tagger (Adler, 2007). This difference is caused by the different ambiguity levels for two languages: level of about 2.7 per token for Hebrew (Goldberg et al., 2009) as opposed to 1.41 for English (Dermatas & Kokkinakis, 1995).

(4) Different gold standard summaries (abstracts vs. extracts) for each corpus. Since the English corpus contains abstracts where humans have used different words from those used in the source text (synonyms, etc.), the words overlap between extract and gold standard keywords is supposed to be less than in the Hebrew corpus with gold standard extracts. Intuitively, we believe that humans describe the most important topics in abstracts by exactly the same words as in the text (think about writing review for some paper that uses term "location method" instead of "position method" – it is more natural to use the same term in the review), therefore the results on the English corpus are more significant than on the Hebrew corpus, where the overlap between the less important common words may affect the results.

Figure 6 helps the reader to compare DegExt performance on English and Hebrew corpora in terms of Precision, Recall, F-measure and Area Under Curve.

Figure 4 presents five keyphrases, original and translated, extracted by TextRank and DegExt, from one of the Hebrew documents entitled "45 years in prison for Gabriel Turgeman, "elusive pedophile"." (translation from Hebrew).[17]

As one can see from the examples, DegExt usually tends to extract bigger phrases than GenEx and TextRank, that is penalizing the evaluation results for DegExt when we consider the whole extracted phrases as output. Figure 7 demonstrates the comparative results for three approaches (DegExt presented

[17] In our corpus it appears under name "doc1.txt".

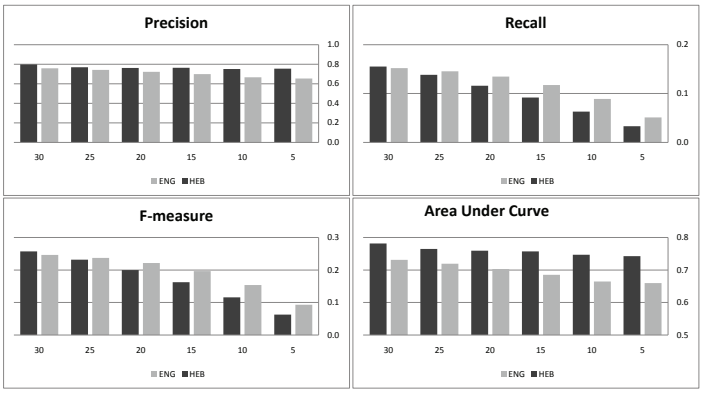

Fig. 6. DegExt: multilingual performance.

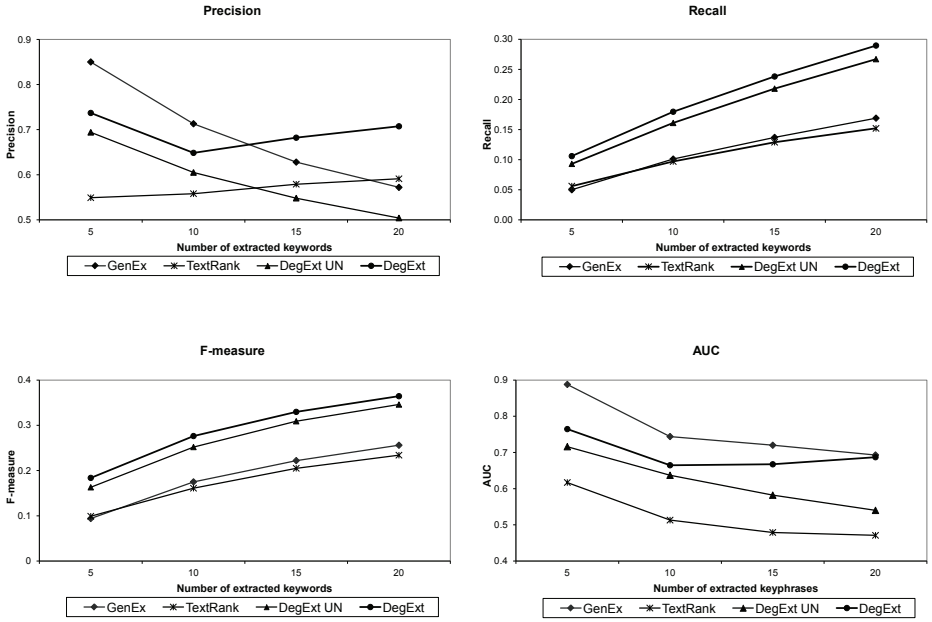

Fig. 7. English: evaluation results for single words and four models respectively (5 - 20 keywords)

with, as it was described in this paper, and without, denoted by DegExt UN, POS filtering) on English corpus, where single words from the extracted keyphrases are considered as output[18]. As it can be seen, DegExt outperforms all other

[18] Of course, true positive in this case is calculated against all single *words* in the gold standard.

approaches in terms of recall and F-measure, also it outperforms TextRank in terms of precision and AUC, when the single extracted words are considered as keyphrases. Also, DegExt outperforms GenEx in terms of precision for the number of keywords greater than 12 and reaches the GenEx AUC score for the number of extracted keywords greater than 20.

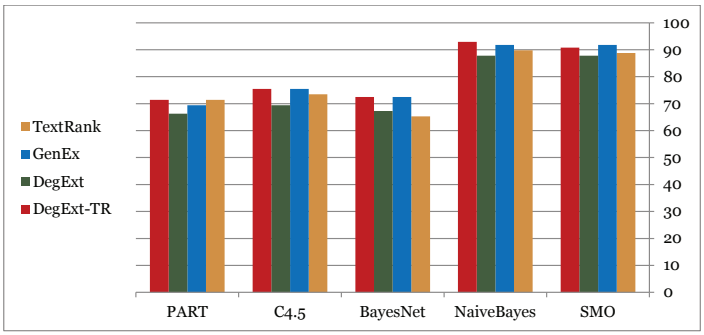

Fig. 8. Classification accuracy for TextRank, GenEx, DegExt, and DegExt-TR (from right to left)

As an additional experiment, we evaluated the relevance of the extracted key phases in the task different from summarization – text categorization. We used extracted key phrases for building the 'bag-of-words' representation of documents, with a maximal size of 30 for a single document. We performed experiments on Yahoo!^{TM} benchmark collection of web documents, called F-series (Moore et al., 1997). The F-series contains 98 HTML documents belonging to one of four major categories: manufacturing, labor, business & finance and electronic communication & networking. Figure 8 demonstrates the resulting accuracy for five different classification algorithms—Decision Tree (C4.5), Decision Rules (PART), Bayesian Network (BayesNet), Naïve Bayes, and Support Vector Machine implemented by a Sequential Minimal Optimization (SMO)—and four keyphrase extraction approaches—TextRank, GenEx, DegExt, and DegExt based on the TextRank representation graphs (DegExt-TR). As can be seen, the hybrid approach (DegExt-TR) applying DegExt (ranking of nodes by a degree centrality) on TextRank representation (undirected graphs of words linked by co-occurrence relationships) outperformed all other approaches (the difference was statistically significant in most cases). Among three original approaches, GenEx has the best accuracy in most classifiers.

As an unsupervised algorithm, DegExt does not require time for training, and its computation time is equal to the time required to build the document representation and to scan the constructed graph for each potential keyphrase. Assuming efficient implementation, DegExt has linear computational complexity relative to the total number of words in a document ($O(n)$) with node sorting taking logarithmic time.

4 Conclusions and Future Work

In this paper we introduced DegExt – a graph-based keyphrase extractor for extractive summarization of text documents. We compared DegExt with two approaches to keyphrase extraction: GenEx and TextRank.

Our empirical results suggest that the supervised GenEx approach has the best precision (a finding that can be explained by using precision as a fitness function of the GA) and AUC for a small number of extracted keyphrases. However, the major disadvantages of this approach are a long training time and language dependency. Inspite of its good performance, GenEx is a supervised learning method with a high computational complexity for the training phase (Turney, 2000), it should be retrained in order to perform well on different types of documents and multiple languages.[19]

When there is no high-quality training set of significant size and a large number of keyphrases (above 15) is needed, we recommend using the unsupervised method based on node degree ranking—DegExt—which provides the best precision and AUC values for a large number of keyphrases in English and the best recall and F-measure results for any number of keyphrases in Hebrew. According to our experimental results, we can extract up to 30 phrases with an average precision above 75%, an average recall above 15%, and an F-measure above 24% in both corpora. When the single extracted keywords are evaluated, DegExt outperforms TextRank and GenEx also in terms of recall and F-measure in English corpus.

The key phrases extracted by a hybrid approach integrating DegExt ranking (based on a degree centrality) with a TextRank document representation (words graph with co-occurrence links) compose the best 'bag-of-words' representation of documents in terms of classification accuracy among four evaluated approaches.

The DegExt approach outperforms the other evaluated approaches—GenEx and TextRank—in terms of implementation simplicity and computational complexity. A major advantage of both TextRank and DegExt over GenEx is their language-independence.

In our future research, we intend to evaluate our method on additional languages and corpora (including the first Gold Standard dataset for keyphrase extraction in Hebrew, to be prepared and published by our group). Also, we intend to explore the use of other kinds of information embedded in our graph representation, like position of the words in the document that may be used to favor the selection of terms with wide distributions instead of locally-important terms, when applying DegExt on the bigger documents, as suggested in (Lee & Baik, 2004). Also, position information can be used for calculating such distance-based node centralities, like *closeness* and *betweenness* centralities – widely used 'node importance' metrics. In addition, other graph representations of documents may be evaluated. The "simple" representation can be extended to many different

[19] The version of GenEx tool that we used cannot be applied to any language except English.

variations like a semantic graph. For corpora of HTML documents, edges may be labeled by section ID (Schenker et al., 2005).

Acknowledgments

We are grateful to our project students Hen Aizenman and Inbal Gobits from Ben-Gurion University for providing the TextRank implementation.

Bibliography

Adler, M. (2007). Hebrew morphological disambiguation: An unsupervised stochastic word-based approach. Phd. Thesis, Ben Gurion University.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, *30*, 107–117.

Dermatas, E., & Kokkinakis, G. (1995). Automatic stochastic tagging of natural language texts. *Computational Linguistics*, *21*, 137–163.

DUC (2002). Document Understanding Conference. http://duc.nist.gov.

Goldberg, Y., Tsarfaty, R., M., A., & Elhadad, M. (2009). Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and em-hmm-based lexical probabilities. *Proceedings of the EACL 2009*. Athens, Greece.

Grineva, M., Grinev, M., & Lizorkin, D. (2009). Extracting key terms from noisy and multitheme documents. *Proceedings of the 18th international conference on World wide web* (pp. 661–670).

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Japan.

Lee, J.-W., & Baik, D.-K. (2004). A model for extracting keywords of document using term frequency and distribution. *Proceedings of CICLING 2004*.

Li, D., Li, S., Li, W., Wang, W., & Qu, W. (2010). A semi-supervised key phrase extraction approach: Learning from title phrases through a document semantic network. *Proceedings of the ACL 2010 Conference Short Papers* (p. 296300). Uppsala, Sweden.

Litvak, M., Kisilevich, S., Keim, D., Lipman, H., Gur, A. B., & Last, M. (2010a). Towards language-independent summarization: A comparative analysis of sentence extraction methods on english and hebrew corpora. *Proceedings of the CLIA Workshop (COLING 2010)* (pp. 20–30). Beijing, China.

Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization* (pp. 17–24).

Litvak, M., Last, M., Aizenman, H., Gobits, I., & Kandel, A. (2011). Degext - a language-independent graph-based keyphrase extractor. *Proceedings of the 7th Atlantic Web Intelligence Conference (AWIC'11)* (pp. 121–130). Fribourg, Switzerland.

Litvak, M., Menahem, M., & Last, M. (2010b). A new approach to improving multilingual summarization using a genetic algorithm. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 927–936). Uppsala, Sweden.

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

Mihalcea, R., & Tarau, P. (2004). Textrank – bringing order into texts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain.

Moore, J., Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., & Mobashe, B. (1997). Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering. *In 7th Workshop on Information Technologies and Systems*.

Schenker, A., Bunke, H., Last, M., & Kandel, A. (2004). Classification of web documents using graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, *18*, 475–496.

Schenker, A., Bunke, H., Last, M., & Kandel, A. (2005). *Graph-theoretic techniques for web content mining*. World Scientific Pub Co Inc.

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL 2003* (pp. 252–259). Edmonton, Canada.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, *2*, 303–336.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). Kea: practical automatic keyphrase extraction. *Proceedings of the fourth ACM conference on Digital libraries* (pp. 254–255). Berkeley, California, USA.