SemanticRank: Ranking Keywords and Sentences Using Semantic Graphs

George Tsatsaronis [1] and **Iraklis Varlamis** [2] and **Kjetil Nørvåg** [1]
[1]Department of Computer and Information Science,
Norwegian University of Science and Technology
[2] Department of Informatics and Telematics, Harokopio University of Athens

Abstract

The selection of the most descriptive terms or passages from text is crucial for several tasks, such as feature extraction and summarization. In the majority of the cases, research works propose the ranking of all candidate keywords or sentences and then select the top-ranked items as features, or as a text summary respectively. Ranking is usually performed using statistical information from text (i.e., frequency of occurrence, inverse document frequency, co-occurrence information). In this paper we present *SemanticRank*, a graph-based ranking algorithm for keyword and sentence extraction from text. The algorithm constructs a *semantic graph* using implicit links, which are based on semantic relatedness between text nodes and consequently ranks nodes using different ranking algorithms. Comparative evaluation against related state of the art methods for keyword and sentence extraction shows that *SemanticRank* performs favorably in previously used data sets.

1 Introduction

Graph based ranking algorithms can be very helpful when searching for important pages in the World Wide Web, members in a social network, or authors in a publication database. Such algorithms capitalize on the existence of explicit links (e.g., hyperlinks, citations) between the graph vertices. In the case of flat text collections, neither links nor citations exist, so the need to devise implicit edges between text keywords or sentences arises. One feasible solution is to exploit the contextual information of terms and create semantic graphs from text based on content similarity. However, conceptual analysis of text has a strong potential in this direction, since it reveals latent similarities between text segments that discuss the same subject but with different terminology. In this direction, we introduce *SemanticRank*, a new algorithm for ranking text segments. *SemanticRank* comprises two steps: (a) creation of semantic graphs from text using both semantic and statistical information, and (b) application of a graph-based ranking algorithm that exploits the edges' weights.

The key contributions of this work are: (1) a novel method for the construction and weighting of the semantic graph, which contains text segments (terms or sentences) as nodes and weighted edges that capture the semantic relatedness (i.e., relatedness in meaning) between nodes but also consider statistical information, (2) the modular design of the method, which allows any graph-based ranking algorithm to be employed, and (3) thorough experimental evaluation of *SemanticRank* in the keyword extraction and text summarization tasks, and evaluation of several alternatives for the graph-based ranking component.

The paper is organized as follows: Section 2 discusses related work. Section 3 presents the preliminaries of *SemanticRank*: the semantic relatedness measure, the graph creation process and the ranking algorithm alternatives. Section 4 provides the details of our method. Section 5 presents the experimental evaluation of *SemanticRank* in two different tasks: keyword extraction and text summarization. Finally, Section 6 concludes and provides pointers to future work.

2 Related Work

This work addresses the problem of extracting the most representative keywords and sentences from text as a means of text summarization. More specificall , *SemanticRank* capitalizes on the creation of term and sentence graphs from text and on graph-based ranking algorithms in order to support the following tasks: (a) keyword extraction from text, which is performed by selecting the top-ranked terms as the most representative ones, and (b) text summarization, which is done by selecting the top ranked sentences as the most representative ones. For this reason, a survey of research works in keyword extraction and text summarization, with emphasis on graph-based approaches is necessary to understand the requirements for these tasks, to locate benchmark data sets, and state of the art graph-based approaches for the comparative evaluation.

2.1 Keyword Extraction

Keyword extraction is an important task in document indexing, and strongly affects the performance of re-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1074–1082,
Beijing, August 2010

trieval, classification clustering, and summarization. Most keyword extraction approaches rely on statistical measures such as term frequency (TF), inverse document frequency (IDF), and variations (Aizawa, 2003). Several works in keyword extraction construct semantic networks from text in order to capture the implicit relations between the individual candidates. Huang et al. (2006) propose the construction of one semantic network per document, and use edges that capture syntactic relations between document terms. Mihalcea and Tarau (2004) suggest a semantic network model where edges express the co-occurrence of terms in the document's sentences. Wang et al. (2007) employ the well known *PageRank* algorithm to perform word sense disambiguation (WSD) and keyword extraction from documents. The graphs that they construct are always subgraphs of the *WordNet* thesaurus[1], which results in low text coverage.

In this work we aim at the design and implementation of a keyword extraction algorithm that takes into account different aspects of text, such as statistical information and the semantic relatedness between keywords. To the best of our knowledge, the current work is the first to propose a semantic network construction model for keyword extraction based on measures of semantic relatedness between keywords. In Section 3 we discuss the employed measure of semantic relatedness, which utilizes both *WordNet* and *Wikipedia*[2] to increase term coverage.

2.2 Text Summarization

The aim of automatic text summarization is to generate a summary of a pre-specified length for a given input text. The *Document Understanding Conference* (DUC) [3] series have provided benchmark data sets with documents and manually generated summaries, which can be used for the evaluation of any automatic text summarizer. Research works in this area conduct automatic text summarization by selecting the most important sentences from the input texts (Steinberger and Jezek, 2009). Baseline methods are based on the observation that important sentences inside a text usually occur at its beginning. Thus, a straightforward baseline is to select the first k sentences as a summary of the text, and setting k in such a way that does not violate the summary length restriction.

Another important class of automated text summarization methods is that of cohesion-based methods. Such methods assume that the important sentences or paragraphs of a given text document are the most connected entities in more or less elaborate semantic structures inside the text. In this direction, the methods construct a graph for each text document, with the vertices being the document sentences, and attempt to determine the most connected vertices in each graph. These methods are further classified depending on how the graph's edges are constructed, for example using word co-occurrences (Salton et al., 1997), local salience and grammatical relations (Boguarev and Kennedy, 1997), co-reference (Baldwin and Morton, 1998), and combinations of the aforementioned (Mani and Bloedorn, 1998).

More recently, some cohesion-based methods have attempted to capture the semantic similarity of sentences inside a text document, and rank sentences in the constructed semantic graph. For example in the method of Mihalcea and Tarau (2004), the graph contains a vertex for each sentence of the given text document, and weighted edges between sentence. The weights represent the semantic similarity between sentences and are actually the contextual overlap between the sentences' terms. The *PageRank* algorithm is then applied to rank sentences in each of the constructed semantic graphs. In the method of Litvak and Last (2008), the vertices are again the sentences of the given text document and the edges represent the syntactic relations between them. Finally, the *HITS* algorithm is applied on the graph for ranking the sentences.

The conclusion from the literature review is that modern trends in graph-based approaches focus on novel methodologies for weighting edges and constructing semantic graphs, and employ standard techniques for ranking vertices, such as *PageRank*, *HITS*, or variations. Another important finding is that the potentiality of creating the edges between the vertices based on measures of semantic relatedness among the respective nodes is unexploited so far, and this is the core of the current work. Thus, the main difference between *SemanticRank* and the aforementioned approaches is that our edge weighting method employs a measure of semantic relatedness between sentences, that is based on *WordNet* and *Wikipedia*. The motivation behind such a perspective is that such semantic graphs would capture the similarity in meaning among the graph vertices, which was neglected by previous approaches. Finally, regarding the data sets used for evaluation, most works in text summarization use past *DUC* data, whereas in the case of keyword extraction a subset from the *Inspec* bibliographic database has been used in several cases in the past (Mihalcea and Tarau, 2004; Hulth, 2003).

[1] http://wordnet.princeton.edu/

[2] http://wikipedia.org

[3] http://duc.nist.gov/. Recently it has been renamed to *Text Understanding Conference* (TAC).

3 Terminology and Preliminaries

A graph-based method for ranking keywords or sentences by constructing semantic graphs comprises two steps: (a) the creation of the semantic graph, with keywords or sentences as vertices, and edges constructed based on a semantic similarity measure between vertices (c.f. Section 3.2), and (b) the adaptation of a new or existent ranking algorithm which analyzes the graph structure and ranks the nodes. Section 3.1 introduces the used terminology. In Section 3.2 we explain how the firs step is done by *SemanticRank*, and in Section 3.3 we present several alternatives for the second step.

3.1 Terminology

In the following we denote with $T(t_i, t_j)$ a pair of terms that occur in text document T. We also assume that T is a member of a document collection D given as input to our method. O represents the used knowledge-base (e.g., thesaurus, dictionary); in our case we are using two such knowledge-bases, namely *WordNet* and *Wikipedia*. With $\mathit{SR}_O(t_i, t_j)$ we denote the semantic relatedness between terms t_i and t_j using O for its computation, and $\mathit{SRS}(A, B)$ represents the semantic relatedness between text segments A and B (e.g., documents, sentences).

Concerning *WordNet*, S_i (S_j) represents the set of the different meanings (senses) with which t_i (t_j) may appear in O. P_{ij} denotes the set of paths connecting senses in S_i with senses in S_j, as these may be found using O. P_{ij}^k represents one such path in the set of paths P_{ij}, namely the k_{th} path. S_{ij} stands for the set of all possible sense pairs between the set of senses S_i and the set of senses S_j. Thus, $|S_{ij}| = |S_i| * |S_j|$. Respectively, S_{ij}^m stands for one such combination, namely the m_{th} combination.

With regards to Wikipedia, W refers to all the *Wikipedia* articles. With a_i we denote the *Wikipedia* article for term t_i. $\mathit{In}(a_i)$ is the set of *Wikipedia* articles that contain at least one link to a_i.

Finally, if d_i is the i_{th} document of D and t_a a term in d_i, then we denote with $\mathit{TF\text{-}IDF}(t_a, d_i) = \frac{\mathit{Count}(t_a, d_i)}{|d_i|} \cdot \log_2 \frac{\mathit{Count}(t_a, D)+1}{|D|}$ the *TF-IDF* weight of t_a in d_i, where $|d_i|$ is the number of term occurrences in d_i, $|D|$ is the number of documents in D, $\mathit{Count}(t_a, d_i)$ the number of occurrences of t_a in d_i, and $\mathit{Count}(t_a, D)$ the number of documents in D that contain t_a.

3.2 Creating Semantic Graphs from Text

A huge volume of literature has been created on how to construct semantic graphs addressing various applications, such as word sense disambiguation (Agirre and Soroa, 2009), keyword and sentence extraction (Mihalcea and Tarau, 2004; Litvak and Last, 2008; Yeh et al., 2008), and computation of semantic relatedness or similarity between terms (Gabrilovich and Markovitch, 2007; Budanitsky and Hirst, 2006; Milne and Witten, 2008).

In this work we adopt a semantic graph construction method which is able to capture the semantic relatedness between terms, as well as text segments. For our purposes we adopt *Omiotis*, the measure proposed by Tsatsaronis et al. (2010) in order to construct and weigh the edges of the semantic graph. *Omiotis* is a knowledge-based measure of semantic relatedness that may capture the semantic relatedness between both keywords and text segments (e.g., sentences, documents), allowing us to construct both semantic keyword graphs for keyword extraction, and semantic sentence graphs for sentence extraction and summarization. Our selection also lies in the fact that *Omiotis* has been shown to perform very well compared to other known measure of semantic relatedness or similarity in tasks such as term-to-term similarity (Tsatsaronis et al., 2010; Budanitsky and Hirst, 2006). However, since *Omiotis* relies solely in *WordNet*, we enhance the coverage of *SemanticRank* by complementing the edge weighting with an additional *Wikipedia*-based measure, namely the measure proposed by Milne and Witten (*MLN*) (2008). In Section 3.2.1 we explain how these two measures are combined in order to compute the semantic relatedness between terms, and in Section 3.2.2 we explain how semantic relatedness is captured between sentences.

3.2.1 Semantic Relatedness Between Terms

The measure presented in (Tsatsaronis et al., 2010) define the semantic relatedness between a pair of terms as shown in Equation 1, where the knowledge-base *O* is *WordNet* (*WN*).

$$\mathit{SR}_{\mathit{WN}}(t_i,t_j)=\max_m\{\max_k\{\mathit{SCM}(S_{ij}^m,P_{ij}^k)\cdot\mathit{SPE}(S_{ij}^m,P_{ij}^k)\}\} \quad (1)$$

where *SCM* and *SPE* are called *Semantic Compactness* and *Semantic Path Elaboration* respectively. Their product measures the weight of the path connecting the two senses in S_{ij}^m, taking into account: the path length, the type of the semantic edges comprising it, and the depth of the intermediate nodes in the *WN* senses hierarchy. The semantic relatedness between two terms t_i, t_j, when $t_i \in \mathit{WN}$ and $t_j \notin \mathit{WN}$, or vice versa, is considered 0. The intuition behind Equation 1 is that the semantic relatedness between two terms should be computed based on the *highest value* path connecting any pair of senses of the two terms. The computation of the *value* takes into account in tandem all of the aforementioned factors.

In order to enhance the coverage of the measure in Equation 1, we combine it with the *WLM Wikipedia*-

based measure of Milne and Witten (2008), which is a low-cost solution for measuring relatedness between terms using the *Wikipedia* articles and link structure as a knowledge base. The semantic relatedness between two terms t_i and t_j according to *WLM* is defined as shown in Equation 2. The intuition behind this formula is that the semantic similarity between two terms becomes higher, as the number of articles pointing to both respective *Wikipedia* articles increases (i.e., as the percentage of the articles linking to both pages compared to the number of articles linking to either of them increases).

$$SR_{Wiki}(t_i,t_j)=\frac{log(\max\{|In(a_i)|,|In(a_j)|\})-log(|In(a_i)\cap In(a_j)|)}{log(|W|)-log(\min\{|In(a_i)|,|In(a_j)|\})} \quad (2)$$

We combine the two measures in a single measure $SRT(t_i, t_j)$, as shown in Equation 3. The reason we prioritize $SR_{WN}(t_i, t_j)$ from $SR_{Wiki}(t_i, t_j)$, when both terms exist in *WN*, is because the former measure has shown much better performance in capturing the semantic relatedness between terms.

$$SRT(t_i, t_j) = \begin{cases} 1, & t_i = t_j \\ SR_{WN}(t_i, t_j), & \text{if } t_i, t_j \in \textit{WordNet} \\ SR_{Wiki}(t_i, t_j), & \text{if } t_i, t_j \in \textit{Wikipedia} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

3.2.2 Semantic Relatedness Between Texts

To quantify the semantic relatedness for a pair of text segments, we build upon the *SRT* measure, but also take into account the statistical importance of the terms occurring in the respective texts. Given two text segments A and B, and two terms $t_a \in A$ and $t_b \in B$, a measure that combines the statistical importance of t_a and t_b, according to (Tsatsaronis et al., 2010), is the harmonic mean of their *TF-IDF* weights. We denote this quantity as λ_{t_a,t_b}. Then for each term $t_a \in A$, we search for the corresponding term $t_b \in B$, which we symbolize with b_*, that maximizes the product of their combined statistical importance and semantic similarity. In our case, b_* is found by Equation 4. Similarly we can find for each $t_b \in B$ the corresponding a_*.

$$b_* = \arg\max_{t_b \in B}\{\lambda_{t_a,t_b} \cdot SRT(t_a, t_b)\} \quad (4)$$

After finding the set of all b_* and a_* terms, the semantic relatedness between the two texts A and B is computed as shown in Equation 5.

$$SRS(A, B) = \frac{\theta(A, B) + \theta(B, A)}{2} \quad (5)$$

where $\theta(A, B) = \frac{1}{|A|}\sum_{t_a \in A} \lambda_{t_a,b_*} \cdot SRT(t_a, b_*)$, and $\theta(B, A)$ can be computed respectively. The measure in Equation 5 is the measure used by *SemanticRank* to construct the edges between sentence vertices in the case of the semantic sentence graphs for text summarization. Regarding which sense of each term is used for the computation of its semantic relatedness with any other term, the senses that maximize the measure in Equation 3 are picked in each case.

3.3 Ranking Nodes in Semantic Graphs

For the purposes of our experimentation we will be evaluating *SemanticRank* with variations of the known *PageRank* and *HITS* algorithms. Some of those variations are applied for the first time in the framework of ranking nodes in semantic graphs. However, as will be explained in Section 4, in *SemanticRank*, any available vertex ranking methodology can be used instead.

The original versions of *PageRank* and *HITS* rely on the *"rich get richer"* model, which is based on explicit links and ignores edges weights. More specifically, HITS prioritizes good hubs and authorities, whereas PageRank uses a dampening factor (β) in order to avoid clique attacks and promote the centrality of nodes. However, in the case of graphs with implicitly devised links, like in semantic graphs, the edges carry weights, which must be taken into account. In this direction, we employ a modified version of the original *PageRank* algorithm, first introduced by Mihalcea and Tarau (2004). The modified *PageRank* is shown in Equation 6.

$$WPR(i) = (1 - \beta) + \beta \cdot \sum_{j \in IN(i)} \frac{w_{ij} \cdot WPR(j)}{\sum_{k \in OUT(j)} w_{jk}} \quad (6)$$

where i, j, k represent vertices, $IN(i)$ and $OUT(j)$ are the sets of *inlink* nodes of i and *outlink* nodes of j respectively, and w_{ij} is the weight of the edge between nodes i and j. In the case of semantic graphs constructed for keyword extraction, nodes are terms, and, thus, $w_{ij} = SRT(t_i, t_j)$. In the case of semantic graphs constructed for text summarization, i and j are sentences, and, thus, $w_{ij} = SRS(i, j)$.

Similarly to the modification shown in Equation 6 for *PageRank*, we can define a weighted version of *HITS*. The respective *authority* and *hub* scores are shown in Equations 7 and 8.

$$authority(i) = \sum_{j \in In(i)} w_{i,j} \cdot hub(j) \quad (7)$$

$$hub(i) = \sum_{j \in Out(i)} w_{i,j} \cdot authority(j) \quad (8)$$

The aforementioned modifications have been already applied in the past in the case of semantic graphs, with application to keyword extraction and text summarization (Mihalcea and Tarau, 2004; Mihalcea, 2004), although using different semantic graphs. For the extraction of the most important nodes, the modified *PageRank* version is used to rank the nodes according to their final *PageRank* values, and the modified *HITS* to rank nodes according to their final *authority* values. In this work, we also consider and evaluate two additional modifications of *PageRank* in order to rank vertices in the case of the semantic keyword graphs. The first modification, that we call *Averaged PageRank Weighting* (*APW*) is presented in Equation 9, and is used after the weighted *PageRank* of Equation 6 has executed. The intuition behind *APW* is that each vertex t_i in the case of the keyword semantic graphs, has a known importance based on its frequency of occurrence (*TF-IDF* weight) inside the given document collection D. Thus, *APW* considers both the importance of vertex t_i inside its semantic graph, and inside its document collection.

$$APW(t_i) = \frac{1}{2}\left(\frac{WPR(t_i)}{WPR_{max}} + \frac{TF\text{-}IDF(t_i, d_j)}{TF\text{-}IDF_{max}}\right) \quad (9)$$

where d_j the specific document from which the semantic keyword graph is created, WPR_{max} is the maximum *PageRank* score found in this graph, and $TF\text{-}IDF_{max}$ is the maximum *TF-IDF* weight found in document d_j.

The second *PageRank* modification that is employed for the first time in the case of semantic keyword graphs is the *priors biased PageRank* (*P-PR*) discussed in (White and Smyth, 2003). The idea is very similar to the works in (Haveliwala, 2002) and (Agirre and Soroa, 2009), and pertain to ranking the nodes in the graph, with regards to a given set of nodes called *priors*. In short, while *PageRank* provides a global ranking of the nodes in the graph, *P-PR* provides a ranking of the nodes with regards to the set of the given *prior* nodes. This is expressed in Equation 10. The only difference with equation 6 is that each node i has its own "*random jump*" probability to the *prior* nodes. Thus, for each node i, *P-PR* has a β_i, which expresses how often we may jump back to the set of the *prior* nodes from node i. The intuition behind *priors* is that certain nodes in the graph are favored against other. In a keyword extraction task the *priors* set may contain the keywords appearing in the document's title.

$$P\text{-}PR(i) = (1-\beta_i)+\beta_i \cdot \sum_{j \in IN(i)} \frac{w_{ij} \cdot P\text{-}PR(j)}{\sum_{k \in OUT(j)} w_{jk}} \quad (10)$$

4 SemanticRank

In this section we present *SemanticRank* (illustrated in Algorithm 1), our algorithm for ranking terms and sentences based on their semantic relatedness. The first step of *SemanticRank* is the semantic graph creation. In the case of semantic keyword graphs, and given a document d_j which belongs in a document collection D, as a preprocessing step, the algorithm detects all $n-$grams of size up to 5 words using a dictionary look-up (i.e., both *WordNet* and *Wikipedia*), and a sliding window, in order to identify candidate keywords, which may be essentially composite terms. The resulting set of terms (i.e., can be terms of 1 to 5 words), which we denote as K_{d_j}, is used for the creation of a graph G with the vertices being all the distinct terms $t_i \in K_{d_j}$. As edge weights w_{ij} *SemanticRank* uses $SRT(t_i, t_j)$ which captures the semantic relatedness between terms t_i and t_j. However, ideally we would also like to incorporate in w_{ij} the statistical information of terms t_i, t_j that we have from

Algorithm 1 SemanticRank(D,*Mode*)

```
INPUT: A text document collection $D$, and a Mode flag
OUTPUT: A ranking $R$ of the semantic graph nodes for every document $d_j \in D$.
Execute(D,Mode)
if Mode is Keywords then
    Identify composite terms of length up to 5 words
end if
Compute and index TF-IDF values for all terms
for all $d_j \in D$ do
    $G$: An initially empty graph
    $G$ = Construct_Semantic_Graph($d_j$,Mode)
    $R$ = Rank_Nodes($G$)
end for
Construct_Semantic_Graph($d_j$,Mode)
$G$: an initially empty graph
if Mode is Keywords then
    Initialize $G$ with $K_{d_j}$
else
    Initialize $G$ with $Sen_{d_j}$
end if
for all pairs of vertices $(v_i, v_j)$ do
    if Mode is Keywords then
        $w_{i,j} = w_{j,i} = \lambda_{v_i,v_j} \cdot SRT(v_i, v_j)$
    else
        $w_{i,j} = w_{j,i} = SRS(v_i, v_j)$
    end if
end for
RETURN $G$
Rank_Nodes(G)
Execute Weighted PageRank in $G$
$R$ = Rank vertices of $G$ in descending order of PageRank values
RETURN $R$ with their PageRank values
```

their frequency of occurrence inside d_j and D. Thus, Equation 11 shows this combination, and it is the formula according to which *SemanticRank* computes the edge weights w_{ij} in the case of the semantic keyword graphs. In the case of semantic sentence graphs creation, *SemanticRank* initializes G with all the distinct sentences Sen_i in d_j as vertices, and it uses Equation 5 to compute the weights between every pair of vertices (i.e., between every pair of sentences). In Algorithm 1 we denote the set of distinct sentences in d_j with Sen_{d_j}.

$$w_{ij} = \lambda_{t_i,t_j} \cdot \mathit{SRT}(t_i, t_j) \qquad (11)$$

In both cases, for the given document d_j, and after the creation of the semantic graph, nodes may be ranked according to the values produced by applying either Equation 6, or Equations 7 and 8. For the case of semantic keyword graphs, the top-k ranked nodes are selected as the most important keywords of d_j. For the case of semantic sentence graphs, the top-k ranked nodes are selected as the set of sentences, put together to constitute the automatically generated summary of d_j. In Algorithm 1 we may substitute line 26 with any of the ranking options discussed in Section 3.3. An analogy can be also drawn with PageRank's *random surfer model*, where a user browses the Web by following links from any given Web page. In the context of text modelling, *SemanticRank* implements what we refer to as *text surfing*, which relates to the concept of text cohesion (Halliday and Hasan, 1976), i.e., from a certain concept in a text, we are likely to *follow* ”links” to related concepts, meaning concepts that have lexical or semantic relation to the current concept.

5 Experimental Evaluation

The experimental evaluation is performed in two tasks: (a) keyword extraction, and (b) text summarization. In both cases we create a semantic graph for each document and we rank the nodes accordingly, using Algorithm 1. For our evaluation we use all the ranking algorithm alternatives described in Section 3.3, and compare results with state of the art approaches that use the same ranking algorithms but different graph creation and edge weighting approaches. The various tested ranking alternatives are: weighted *SemanticRank* (*Sem*) using *PageRank* (*WPR*), and *HITS* (*WHITS*), and unweighted *SemanticRank* (*USem*) using the original versions of *PageRank* (*UPR*) and *HITS* (*UHITS*). In the case of keyword extraction we evaluate additionally the *Averaged PageRank Weighting* (*APW*) and *PageRank Priors* (*PPR*), where the *prior* nodes were set to the terms occurring in each abstract's title.

Method		P	R	F
Sem (k=5)	WPR	0.396	0.121	0.1853
	WHITS	0.348	0.088	0.14
	APW	0.556	0.185	0.278
	P-PR	**0.659**	0.226	0.337
Sem (k=10)	WPR	0.368	0.2463	0.296
	WHITS	0.335	0.138	0.195
	APW	0.498	0.331	0.398
	P-PR	0.524	0.352	0.422
Sem (k=15)	WPR	0.371	0.364	0.368
	WHITS	0.355	0.241	0.287
	APW	0.449	0.442	0.446
	P-PR	0.451	0.441	0.446
Sem (k=20)	WPR	0.376	0.466	0.417
	WHITS	0.374	0.312	0.34
	APW	0.421	**0.532**	**0.47**
	P-PR	0.418	0.514	0.46
USem (k=5)	UPR	0.057	0.046	0.048
	UHITS	0.061	0.053	0.055
USem (k=10)	UPR	0.06	0.102	0.07
	UHITS	0.06	0.108	0.072
USem (k=15)	UPR	0.052	0.116	0.069
	UHITS	0.054	0.123	0.072
USem (k=20)	UPR	0.052	0.14	0.074
	UHITS	0.053	0.151	0.076
Michalcea (2004)		0.312	0.431	0.362
Hulth (2003)		0.252	0.517	0.339

Table 1: Results of the keyword extraction task in the Inspec database.

5.1 Keyword Extraction

We applied *SemanticRank* in an automated keyword extraction task on the Inspec database[4]. The Inspec database stores abstracts of journal papers from computer science and information technology and the keyword extraction task aims in selecting the most descriptive keywords for each abstract. Each abstract has been already assigned keywords by professional indexers, which constitute the gold standards for systems' comparison. The mean number of assigned terms per abstract from the experts is 7.63. The goal is to extract as many of the keywords suggested by the professional indexers as possible for each abstract. In this data set our results are directly comparable to the works in (Mihalcea and Tarau, 2004) and (Hulth, 2003).

We evaluate *SemanticRank* (*Sem*) using varying k values ($5, 10, 15$, and 20), where k stands for the number of keywords to be extracted from each abstract. In Table 1 we report the results of macro-averaged pre-

[4] Many thanks to Anette Hulth for providing us the data set used in her keyword extraction experiments.

System		F-Measure
Sem	WPR	0.40996(0.39067 − 0.4292)
	WHITS	0.3651(0.3435 − 0.38609)
USem	UPR	0.2951(0.2727 − 0.3195)
	UHITS	0.3132(0.2901 − 0.3375)
T		0.4131(0.3922 − 0.434)
P		0.4039(0.3843 − 0.4226)
O		0.3905(0.3663 − 0.4132)
V		0.3885(0.368 − 0.4085)
Q		0.3857(0.3616 − 0.4089)
Baseline		0.3549(0.3329 − 0.3756)

Table 2: Results (F-Measure) of the single-document summarization task, (DUC 2001).

System		F-Measure
Sem	WPR	0.4971(0.4799 − 0.5164)
	WHITS	0.3836(0.3815 − 0.4047)
USem	UPR	0.3086(0.297-0.32084)
	UHITS	0.2851(0.2735-0.297)
TextRank		0.4904
S27		0.5011
S31		0.4914
S28		0.489
S21		0.4869
S29		0.4681
Baseline		0.4779

Table 3: Results (F-Measure) of the single-document summarization task, (DUC 2002).

cision (P), recall (R), and F-Measure (F) over all abstracts. Precision for each abstract is the number of correctly extracted keywords, divided by the number of extracted keywords, and recall differs only in the denominator (number of keywords suggested by the indexers). We also present the best reported results for the algorithms in (Mihalcea and Tarau, 2004), and (Hulth, 2003).

Results show that *SemanticRank* with weighted *PageRank* gives better F-Measure from the approaches in (Mihalcea and Tarau, 2004) and (Hulth, 2003) for $k = 15$ and $k = 20$ and always better from weighted *HITS*. *APW* and *P-PR* have higher F-Measure than *WPR*, achieving top performance (bold values), with *APW* producing the best F-Measure for $k = 20$. In this case, the difference between *APW* and *TextRank*, both in precision and recall, was found statistically significant at the 0.95 confidence level, using Fisher's exact test. In addition, we can observe that the unweighted versions of *PageRank* and *HITS* produce very poor results. This shows that our method benefits greatly from the suggested edges' weighing scheme.

5.2 Text Summarization

We evaluated *SemanticRank* in two different text summarization tasks: single-document, and multi-document summarization. As in the keyword extraction task, we evaluate both the weighted and the unweighted versions of *SemanticRank* (*Sem* and *USem*) using *WPR*, *WHITS*, *UPR*, and *UHITS* respectively. We also compare against state of the art results in the used data sets, and we report on results from related methods (i.e., *TextRank*) when possible.

5.2.1 Single Document Summarization

In the single-document summarization task we have used the data sets of the *Document Understanding Conference* (*DUC*) from the 2001 and 2002 competitions. The two data sets comprise 308 and 567 news articles respectively. For both data sets, two reference summaries per document were provided. The task for the participating systems in both competitions was to provide for each document a summary of at most 100 words. Thus, we apply *SemanticRank* by first ranking sentences following Algorithm 1, and then by merging them, starting from the top ranked sentences, until the 100 words limit is reached. For the evaluation against the reference summaries, we are using the *ROUGE* toolkit, which is based on $N-$grams, and has been the standard evaluation methodology for the summarization task (Lin and Hovy, 2003) in all the recent DUC competitions. Since in *DUC* 2001 and *DUC* 2002 the *ROUGE* system was not the standard evaluation toolkit, we implemented the evaluation of the two tasks in *ROUGE*. The setup we adopted for ROUGE was (*Ngram(1,1)*, stemmed words and no stopwords), identical to the one adopted in (Mihalcea and Tarau, 2004).

In Table 2 we present the F-Measure values produced from *ROUGE* for *SemanticRank*, and the top 5 performing systems (participating systems *T*, *P*, *O*, *V*, and *Q*), for the 2001 data set. Similarly, Table 3 presents the results for the 2002 data set. In both cases we report the performance of a simple baseline method, that takes the first sentences from each article, until the limit of 100 words is reached. When available, we also present the results from (Mihalcea and Tarau, 2004), and also the 0.95 confidence intervals for the F-Measure values, as these were generated by *ROUGE*. The results in the two tables show that *SemanticRank*, when the weighted version of PageRank is used, produces very high F-Measure score. In both cases, our system ranks among the top 2 systems in the task.

System		F (R-2)	F (R-SU4)
Sem	WPR	0.093	0.133
	WHITS	0.078	0.115
USem	UPR	0.031	0.069
	UHITS	0.028	0.062
S40		0.111	0.143
S55		0.098	0.135
S45		0.096	0.132
S44		0.093	0.136
S47		0.093	0.130
Baseline		0.085	0.122

Table 4: Results of the multi-document summarization task (DUC 2007 update task).

5.2.2 Multi Document Summarization

For the multi document summarization task we used the data from the *DUC 2007 update task*. The data set consists of 250 documents organized in topics, and each topic is further divided into three clusters, for each of which gold standard summaries are provided by evaluators. In this case, the average of *ROUGE-2* and *ROUGE-SU4* scores are used for evaluation. Table 4 presents the average F-Measure values for both scores. We also report the top-5 performing systems in the respective task, as well as the performance of the generic baseline that was used in this case. As Table 4 shows, the combination of *SemanticRank* with the weighted *PageRank* produces better results than weighted *HITS* and the unweighted versions. This drop in performance compared to the results in the single-document summarization task can be partly explained by the fact that in this case the *DUC 2007 update task* allows for the system to assume previous knowledge for the document clusters B and C of each topic. In our case, we have not embedded any methodology that takes advantage of this knowledge.

Regarding the top system in Table 4, system *S40*, is the system called *GISTEXTER* (Hickl et al., 2007). *GISTEXTER* uses textual inference and textual contradiction to construct representations of knowledge encoded in a document collection. The system comprises four components: question processing, sentence retrieval, sentence ranking, and summary generation. However, for the summary generation component, a set of heuristics is used to generate the summary. In a similar approach (Amini and Usunier, 2007), where coherent text fragments are sought with regards to the initial question, the authors show that query expansion using a contextual approach may lead to fin important terms for the summary, among different related documents. Their system ranked among the top in the main task of *DUC 2007*, leading to the conclusion that for a multi-document summarization system, a contextual approach might be more efficien than *SemanticRank*.

However, from the results presented in Tables 2, 3 and 4, we experienced a very good performance of *SemanticRank* in ranking sentences for the text summarization task, with the weighted ranking variations producing always better results than the unweighted.

6 Conclusions and Future Work

In this paper we introduced *SemanticRank*, a new algorithm for ranking keywords and text segments using measures of semantic relatedness. The novelty of the algorithm is its semantic graph creation step, which is based on a measure of semantic relatedness that combines *WordNet* and *Wikipedia*. We evaluated *SemanticRank* using several alternatives for its ranking step, all based on weighted and unweighted variations of *PageRank* and *HITS*. Results in keyword extraction and text summarization experiments show that it performs favorably over state of the art related methods, and that the selected edges' weighting boosts its performance. In our future work we will examine the potentiality of more graph-based ranking methods, and it is on our next plans to embed *SemanticRank* on more linguistic tasks, such as sentiment analysis and opinion mining.

References

Agirre, Eneko and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proc. of the 12th Conference of the European Chapter of the ACL*, pages 33–41.

Aizawa, Akiko N. 2003. An information-theoretic perspective of tf-idf measures. *Information Processing Management*, 39(1):45–65.

Amini, Massih and Nicolas Usunier. 2007. A contextual query expansion approach by term clustering for robust text summarization. In *Proc. of the DUC 2007 Conference*.

Baldwin, Breck and Thomas S. Morton. 1998. Dynamic coreference-based summarization. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*.

Boguarev, Branimir and Christopher Kennedy. 1997. Salience-based content characterization of text documents. In *Proc. of ACL/EACL Workshop on Intelligent Scalable Text Summarization*.

Budanitsky, Alexander and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Gabrilovich, Evgeniy and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611.

Halliday, Michael A.K. and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman.

Haveliwala, Taher H. 2002. Topic-sensitive pagerank. In *Proc. of the World Wide Web Conference*, pages 517–526.

Hickl, Andrew, Kirk Roberts, and Finley Lacatusu. 2007. Lcc's gistexter at duc 2007: Machine reading for update summarization. In *Proc. of the DUC 2007 Conference*.

Huang, Chong, Yong Hong Tian, Zhi Zhou, Charles X. Ling, and Tiejun Huang. 2006. Keyphrase extraction using semantic networks structure analysis. In *Proc. of the 6th International Conference on Data Mining*, pages 275–284.

Hulth, Anette. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 216–223.

Lin, Chin-Yew and Eduard H. Hovy. 2003. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *Proc. of the Human Language Technology Conference and the North American Chapter of the ACL Conference*.

Litvak, Marina and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proc. of the ACL Workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24.

Mani, Interjeet and Eric Bloedorn. 1998. Machine learning of generic and user-focused summarization. In *Proc. of the 15th National Conference on A.I. and 10th Innovative Applications of A.I. Conference*, pages 821–826.

Mihalcea, Rada and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Mihalcea, Rada. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proc. of the 42nd Annual Meeting of the ACL*.

Milne, David N. and Ian H. Witten. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. of the 1st AAAI Workshop on Wikipedia and Artificial Intelligence*.

Salton, Gerard, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing Management*, 33(2):193–207.

Steinberger, Josef and Karel Jezek. 2009. Text summarization: An old challenge and new approaches. In *Foundations of Computational Intelligence (6)*, pages 127–149.

Tsatsaronis, George, Iraklis Varlamis, and Michalis Vazirgiannis. 2010. Text relatedness based on a word thesaurus. *Journal of Artificial Intelligence Research*, 37:1–39.

Wang, Jinghua, Jianyi Liu, and Cong Wang. 2007. Keyword extraction based on pagerank. In *Proc. of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 857–864.

White, Scott and Padhraic Smyth. 2003. Algorithms for estimating relative importance in networks. In *Proc. of the 9th International Conference on Knowledge Discovery and Data Mining*, pages 266–275.

Yeh, Jen-Yuan, Hao-Ren Ke, and Wei-Pang Yang. 2008. iSpreadRank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Syst. Appl.*, 35(3):1451–1462.