An Ontology-based Approach to Text Summarization

Leonhard Hennig
DAI Labor, TU Berlin
Berlin, Germany
leonhard.hennig@dai-labor.de

Winfried Umbrath
DAI Labor, TU Berlin
Berlin, Germany
winfried.umbrath@dai-labor.de

Robert Wetzker
DAI Labor, TU Berlin
Berlin, Germany
robert.wetzker@dai-labor.de

Abstract

Extractive text summarization aims to create a condensed version of one or more source documents by selecting the most informative sentences. Research in text summarization has therefore often focused on measures of the usefulness of sentences for a summary. We present an approach to sentence extraction that maps sentences to nodes of a hierarchical ontology. By considering ontology attributes we are able to improve the semantic representation of a sentence's information content. The classifier that maps sentences to the taxonomy is trained using search engines and is therefore very flexible and not bound to a specific domain. In our experiments, we train an SVM classifier to identify summary sentences using ontology-based sentence features. Our experimental results show that the ontology-based extraction of sentences outperforms baseline classifiers, leading to higher Rouge scores of summary extracts.

1 Introduction

Text summarization is the task of creating a document from one or more textual sources that is smaller in size but retains some or most of the information contained in the original sources. What information and which other characteristics of the source documents are kept depends on the intended use of the summary [5, 7]. Ultimately, the goal of automatic text summarization is to create summaries that are similar to human-created abstracts. Since this is a challenging task that involves text analysis, text understanding, the use of domain knowledge and natural language generation, research in automatic text summarization has largely focussed on generating extractive summaries [4]. Extracts are summaries that consist of textual units selected from source documents, based on their usefulness for a summary. This usefulness is often equated with salience, hence most approaches evaluate which properties of textual units determine key information that therefore should be contained in a summary. Proposed features include document structure and term prominence [8], rhetorical structure [9], as well as graph-theoretic [2, 10] and semantic features [5].

In [12], knowledge from WordNet[1] as well as from UMLS[2], a medical ontology, is shown to improve the performance of extractive summarization. The authors utilize the ontologies for query expansion, as well as for representing sentences by bag-of-words. The bag-of-words of a sentence contains only those words corresponding to concepts of the ontology. As opposed to our approach, the authors do not use the ontology for computing a similarity measure, but only apply the ontologies as dictionaries of valid concepts. The term-based mapping of sentences to ontologies was also proposed by [3]. The authors exploit the concept generalization options offered by WordNet relations to find the most informative concepts contained in a text.

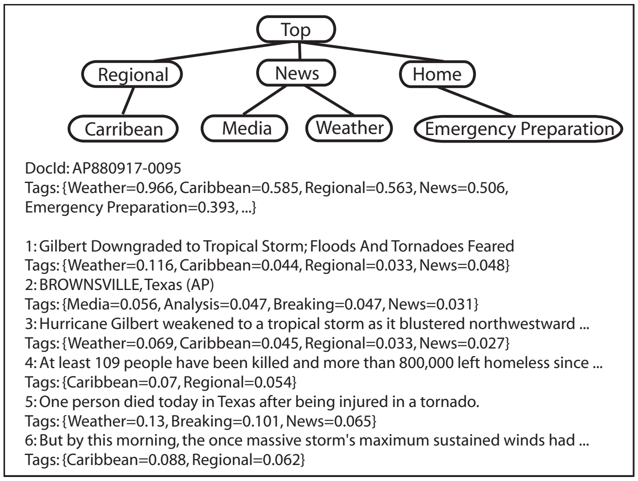
Closely related to our approach is the method proposed by [13]. The authors manually construct an ontology for a small domain of news articles, using the category labels of the ontology tree to score paragraphs. The category's score is increased along with the score of its parent categories if a paragraph contains a category label. The categories with the highest score are then selected as the main topics of the source document and each paragraph is rated based on these topics. Paragraphs are selected until the desired summary length is reached. This approach therefore only ranks paragraphs, not sentences, and only considers the category labels themselves, as even synonyms are not recognized. A more complete overview of automatic text summarization is provided by [4].

Our contribution

We classify sentences to nodes of a predefined hierarchical ontology, i.e. a taxonomy. Each ontology node is populated by a bag-of-words constructed from a web search. Sentences are represented by subtrees in the ontology-space, which allows us to apply similarity measures in the

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

[2] http://www.nlm.nih.gov/research/umls



DocId: AP880917-0095
Tags: {Weather=0.966, Caribbean=0.585, Regional=0.563, News=0.506,
Emergency Preparation=0.393, ...}

1: Gilbert Downgraded to Tropical Storm; Floods And Tornadoes Feared
Tags: {Weather=0.116, Caribbean=0.044, Regional=0.033, News=0.048}
2: BROWNSVILLE, Texas (AP)
Tags: {Media=0.056, Analysis=0.047, Breaking=0.047, News=0.031}
3: Hurricane Gilbert weakened to a tropical storm as it blustered northwestward ...
Tags: {Weather=0.069, Caribbean=0.045, Regional=0.033, News=0.027}
4: At least 109 people have been killed and more than 800,000 left homeless since ...
Tags: {Caribbean=0.07, Regional=0.054}
5: One person died today in Texas after being injured in a tornado.
Tags: {Weather=0.13, Breaking=0.101, News=0.065}
6: But by this morning, the once massive storm's maximum sustained winds had ...
Tags: {Caribbean=0.088, Regional=0.062}

Figure 1. Illustration of DMOZ Categories

ontology-space and to compute relations between sentences based on graph properties of the subtrees. Furthermore, node confidence weights computed by the classifier enable us to identify the main topics of a document.

The ontology we use is not domain-specific. Also, the hierarchical classifier which maps sentences to nodes does not require labelled data during the training phase. The category labels assigned to a sentence by the classifier are employed to compute typical thematic similarity measures as well as graph-based measures, such as subtree overlap. In addition, we exploit the structural information provided by the taxonomy to compute a sentence's specificity based on the depth and number of assigned subtrees.

2 Populating the taxonomy

We use the taxonomy and the hierarchical classifier[3] described in [1]. For our experiments, we create a taxonomy from the topics of the Open Directory Project[4] category tree. The first level "World" branch is excluded, as it consists mostly of non-english content. For computational reasons, we restrict the taxonomy to the first two levels of the DMOZ category tree in our initial experiments, yielding a taxonomy consisting of 1036 nodes. Nodes are represented as *tfidf*-weighted bag-of-words created by extracting and processing all terms from the first 20 websites returned from a YAHOO! search engine query. The search query is constructed from the label assigned to each category and the label of the parent category, e.g. for the node "Artificial Intelligence" in the "Computers" branch the query string would be "Computers Artificial Intelligence". The highest-ranked websites returned by the web service are processed by removing HTML tags, removing stop words, and stemming terms with the Porter algorithm [11].

The feature vector for each category is created by aggregating the feature vectors of the retrieved documents and normalizing the resulting vector to unit length. To represent structural information of the taxonomy, we propagate feature distributions from leaf nodes to parent nodes by recursively aggregating children feature weights to the parent:

$$w_c^{'}(t) = w_c(t) + \sum_{i \in children(c)} w_i(t), \tag{1}$$

where $w_i(t)$ is the *tfidf* weight of term t in child category i.

3 The hierarchical classifier

The classifier maps a sentence to the taxonomy by choosing the subtrees which best represent the sentence. As the distance measure, we compute the cosine distance between the feature vectors of the sentence and a category. The levelwise classification follows the method presented in [1]. Starting at the root node, the algorithm computes the similarity of a sentence to all child nodes, then determines the mean μ and standard deviation σ of the resulting similarities, and selects all nodes for further exploration whose similarity to the sentence $sim(sentence, node) > \mu + \alpha\sigma$. The parameter α determines the branching behaviour. Setting it to a very high value makes the algorithm choose only a single path. If the maximum similarity of a child is lower than the current node's similarity to the sentence, or if a leaf node has been reached, the algorithm stops. A sentence is therefore not necessarily classified to a leaf node, but may be assigned to an internal node.

The classifier assigns all categories of the subtrees as valid tags, thus allowing us to match nodes sharing common subtrees. Figure 1 shows a sample illustration of the DMOZ category tree, as well as the tags assigned by the classifier to sentences of a newspaper article.

4 Creating Ontology-based features

For our ontology-based summarizer we compute a set of features for each sentence based on the output of the hierarchical classifier. The α-parameter controlling the number of tags assigned to a sentence is set to 1.5. We create a bag-of-tags for each sentence by collecting the nodes computed by the hierarchical classifier. If a sentence is mapped to multiple subtrees in the taxonomy, we include all nodes from every subtree. For example, in Figure 1 the first sentence is assigned the subtrees *News/Weather* and *Regional/Carribean*.

We use the classifier's confidence weights to compute a subtree overlap measure for each sentence. By aggregating

[3] Available from http://project.askspree.de/

[4] http://www.dmoz.org

Feature	Description
Tag overlap	Cosine distance of confidence-weighted sentence tag vector to document tag vector
Subtree depth	Depth of the most specific node assigned by the hierarchical classifier
Subtree count	Number of subtrees assigned by the hierarchical classifier

Table 1. Ontology-based features used for extractive summarization

the bag-of-tags of the sentences we can form a document's bag-of-tags:

$$w_d(t) = \sum_{i \in sentences(d)} conf(t, i), \qquad (2)$$

where $w_d(t)$ is the document weight of tag t and $conf(t, i)$ is the confidence value of tag t in sentence i. The tags with the highest weights can be interpreted as the main topics of a document, as can be seen in Figure 1.

We normalize the bag-of-tags with associated confidence values for each sentence and document to unit length. The tag-based similarity measure of a sentence to its document is then the dot product of the two vectors. This measure captures how well a sentence represents the information content of its document in the ontology-space. The number of subtrees computed by the classifier, as well as the tree depth of its most specific tag, are also assigned as features for each sentence. The latter is interpreted as a measure of the specificity of a sentence: If a sentence is classified as a leaf node of a certain depth, it is assumed to contain more specific information than a sentence that is classified to a higher-ranked internal node. The number of subtrees can be interpreted as a measure of the quantity of a sentence's information content. Table 1 lists the features derived from the taxonomy.

5 Extractive summarization

For the evaluation of our approach, we implement a baseline summarizer using well-known features from summarization research. We create a bag-of-words for each sentence by removing stop words and applying stemming. We then compute the average *tf* and *tfidf* scores of the bag-of-words of each sentence as proposed by [8]. Furthermore, we determine the cosine similarity of a sentence to its document. The document's bag-of-words is constructed similarly to the bag-of-tags by aggregating the $tfidf$-weighted bag-of-words of its sentences.

We also compute features based on the structure of a document by assigning each sentence binary features that indicate whether the sentence occurs in the first, second or last third of the document, as well as a real-valued position score in $[0, 1]$, with the first sentence of a document having a score of 1. To prefer sentences of medium length, we assign binary features indicating whether a sentence is shorter respectively longer than some fixed thresholds. For our experiments, we set the minimum length threshold of a sentence to 4 words, and the maximum length threshold to 15, counting only content words.

Furthermore, we implement a baseline summarizer that uses lead sentences. This summarizer simply selects the first sentences from each document in a topic's document collection until the desired summary length is reached.

For the evaluation of our approach we use the DUC 2002 corpus[5], as it provides human-created extracts to train a supervised classifier. The corpus contains 59 data sets of newspaper articles, each set sharing a common topic. We create automatic summaries based on the documents annotated with sentence information to avoid differing sentence splits as a source of error.

Experiments and Results

We train a linear Support Vector Machine (SVM) to classify sentences for extractive summarization. We label all sentences from the human created extracts as positive examples, and all other sentences as negative examples. The sentence feature vectors are scaled such that all feature values are in $[0, 1]$. We set the SVM parameter C controlling the trade off between fit to the training data and model generalization to 2 and parameter J, weighing training errors on positive examples, to 6, aiming at a higher recall of human-extracted sentences, as suggested by [5].

We perform leave-one-out cross validation, with sentences from each topic in turn constituting the test set, and all other sentences being used as training data. The summary is constructed from sentences of the test topic. We order sentences by the classification value computed by the SVM and extract all sentences that are classified as positive instances by the SVM until the desired summary length is reached. We then reorder sentences by their position in the source documents.

From the SVM output we collect per-topic as well as per-sentence precision and recall to compute macro- (per-topic) and micro-averaged (per-sentence) precision, recall and F_1 measures. We also compute the ROUGE measure for the coverage of the automatic summary with respect to the human model extracts. ROUGE is recall-oriented, based on n-gram overlap, and correlates well with human evaluations [6]. The DUC data set provides human model extracts for summaries of length 200 and 400 words.

[5] http://duc.nist.gov

Features	avg. by	Prec.	Rec.	F1
Baseline	macro	0.294	0.304	0.299
	micro	0.266	0.287	0.276
Ontology	macro	0.274	0.380	**0.319**
	micro	0.250	0.362	**0.296**

Table 2. Micro- and macroaveraged precision, recall and F_1 measures for the baseline and the ontology feature set

Feature Set	Rouge-1 F1		Rouge-2 F1	
Length	200	400	200	400
Lead	0.3986	0.5103	0.1604	0.2488
Baseline	0.4386	0.5325	0.1657	0.2632
Ontology	**0.4636**	**0.5716**	**0.2040**	**0.3143**
Best DUC 02	0.50583	0.59064	0.26765	0.34972

Table 3. Comparison of macroaveraged Rouge Scores for different summary lengths

Table 2 compares the micro- and macro-averaged sentence results of the SVM classifier. We compare the baseline features to an extended feature set containing the ontology features. As can be seen, adding the features computed from the ontology improves both micro- and macro-averaged F_1 scores. Microaveraged scores are lower than macroaveraged scores for both feature sets. The ontology-based features seem to support recall, while lowering precision.

Table 3 compares the Rouge scores for summaries with lengths of 200 and 400 words. For both summary lengths, Rouge scores increase when adding the ontology-based features. We have also included the results of a baseline system using only lead sentences for comparison, as well as the results of the best system from DUC 2002. Although our results are not as good as the best system from the DUC 2002 challenge, we note that our hierarchical classifier is not trained on the DUC data. It trades flexibility and generality for accuracy. Also, since it is trained offline, the actual mapping of sentences to nodes of the ontology is very efficient, which is important for an online summarization system.

6 Conclusion

In this paper, we describe how sentences can be mapped to nodes of a flexible, wide-coverage ontology. We show that the mapping provides a semantic representation of the information content of sentences that improves summarization quality. From the category labels themselves as well as from the structural properties of the taxonomy we compute various sentence features which improve the accuracy of an SVM classifier trained on the task of sentence classification. Furthermore, we provide experimental results which show that Rouge scores of summaries generated from the classification output of an SVM trained with ontology-based sentence features outperform summaries generated from an SVM trained only on standard features from summarization research.

Even though the ontology we have used is rather small, the improvement in results seems very promising. We therefore plan to extend our experiments to a larger ontology in order to further test the effectiveness of the proposed features. Another interesting research direction is the transfer of our approach to the more general case of a non-hierarchical ontology.

References

[1] T. Alpcan, C. Bauckhage, and S. Agarwal. An Efficient Ontology-Based Expert Peering System. In *Proc. IAPR Workshop on Graph-based Representations*, pages 273–282, 2007.

[2] G. Erkan and D. Radev. Lexrank: graph-based centrality as salience in text summarisation. *J. of Artificial Intelligence Research*, 2004.

[3] E. Hovy and C.-Y. Lin. Automated text summarization in summarist. In I. Mani and M. Maybury, editors, *Advances in Automated Text Summarization*, pages 18–94. MIT Press, 1999.

[4] K. S. Jones. Automatic summarising: The state of the art. *Inf. Process. Manage.*, 43(6):1449–1481, 2007.

[5] J. Leskovec, N. Milic-Frayling, and M. Grobelnik. Impact of linguistic analysis on the semantic graph coverage and learning of document extracts. In *Proc. of AAAI'05*, 2005.

[6] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proc. of NAACL-HLT 2003*, pages 71–78, 2003.

[7] I. Mani. *Automatic summarization*. John Benjamins Publishing Company, 2001.

[8] I. Mani and E. Bloedorn. Machine learning of generic and user-focused summarization. In *Proc. of AAAI '98/IAAI '98*, pages 820–826, 1998.

[9] D. Marcu. The rhetorical parsing of natural language texts. In *Proc. of the35th Annual Meeting of the ACL*, pages 96–103, 1997.

[10] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proc. of the ACL 2004 on Interactive poster and demonstration sessions*, page 20, 2004.

[11] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[12] R. Verma, P. Chen, and W. Lu. A semantic free-text summarization system using ontology knowledge. In *Proc. of the 2007 Document Understanding Conf. (DUC 07)*, 2007.

[13] C.-W. Wu and C.-L. Liu. Ontology-based text summarization for business news articles. In *Computers and Their Applications 2003*, pages 389–392, 2003.