Automatic single-document key fact extraction from newswire articles

Kastner, I.; Monz, C.

DOI
10.3115/1609067.1609113

Publication date
2009

Document Version
Final published version

Published in
Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics

Automatic Single-Document Key Fact Extraction from Newswire Articles

Itamar Kastner
Department of Computer Science
Queen Mary, University of London, UK
itk1@dcs.qmul.ac.uk

Christof Monz
ISLA, University of Amsterdam
Amsterdam, The Netherlands
christof@science.uva.nl

Abstract

This paper addresses the problem of extracting the most important facts from a news article. Our approach uses syntactic, semantic, and general statistical features to identify the most important sentences in a document. The importance of the individual features is estimated using generalized iterative scaling methods trained on an annotated newswire corpus. The performance of our approach is evaluated against 300 unseen news articles and shows that use of these features results in statistically significant improvements over a provenly robust baseline, as measured using metrics such as precision, recall and ROUGE.

1 Introduction

The increasing amount of information that is available to both professional users (such as journalists, financial analysts and intelligence analysts) and lay users has called for methods condensing information, in order to make the most important content stand out. Several methods have been proposed over the last two decades, among which keyword extraction and summarization are the most prominent ones. Keyword extraction aims to identify the most relevant words or phrases in a document, e.g., (Witten et al., 1999), while summarization aims to provide a short (commonly 100 words), coherent full-text summary of the document, e.g., (McKeown et al., 1999). Key fact extraction falls in between key word extraction and summarization. Here, the challenge is to identify the most relevant facts in a document, but not necessarily in a coherent full-text form as is done in summarization.

Evidence of the usefulness of key fact extraction is CNN's web site which since 2006 has most of its news articles preceded by a list of story highlights, see Figure 1. The advantage of the news highlights as opposed to full-text summaries is that they are much 'easier on the eye' and are better suited for quick skimming.

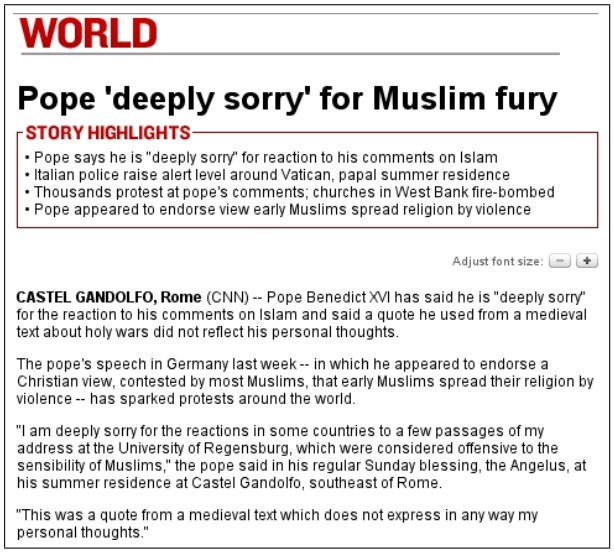
So far, only CNN.com offers this service and we are interested in finding out to what extent it can be automated and thus applied to any newswire source. Although these highlights could be easily generated by the respective journalists, many news organization shy away from introducing an additional manual stage into the workflow, where pushback times of minutes are considered unacceptable in an extremely competitive news business which competes in terms of seconds rather than minutes. Automating highlight generation can help eliminate those delays.

Journalistic training emphasizes that news articles should contain the most important information in the beginning, while less important information, such as background or additional details, appears further down in the article. This is also the main reason why most summarization systems applied to news articles do not outperform a simple baseline that just uses the first 100 words of an article (Svore et al., 2007; Nenkova, 2005).

On the other hand, most of CNN's story highlights are *not* taken from the beginning of the articles. In fact, more than 50% of the highlights stem from sentences that are not among the first 100 words of the articles. This makes identifying story highlights a much more challenging task than single-document summarization in the news domain.

In order to automate story highlight identification we automatically extract syntactic, semantic,

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 415–423,
Athens, Greece, 30 March – 3 April 2009.



WORLD

Pope 'deeply sorry' for Muslim fury

STORY HIGHLIGHTS

- Pope says he is "deeply sorry" for reaction to his comments on Islam
- Italian police raise alert level around Vatican, papal summer residence
- Thousands protest at pope's comments; churches in West Bank fire-bombed
- Pope appeared to endorse view early Muslims spread religion by violence

Adjust font size:

CASTEL GANDOLFO, Rome (CNN) -- Pope Benedict XVI has said he is "deeply sorry" for the reaction to his comments on Islam and said a quote he used from a medieval text about holy wars did not reflect his personal thoughts.

The pope's speech in Germany last week -- in which he appeared to endorse a Christian view, contested by most Muslims, that early Muslims spread their religion by violence -- has sparked protests around the world.

"I am deeply sorry for the reactions in some countries to a few passages of my address at the University of Regensburg, which were considered offensive to the sensibility of Muslims," the pope said in his regular Sunday blessing, the Angelus, at his summer residence at Castel Gandolfo, southeast of Rome.

"This was a quote from a medieval text which does not express in any way my personal thoughts."

Figure 1: CNN.com screen shot of a story excerpt with highlights.

and purely statistical features from the document. The weights of the features are estimated using machine learning techniques, trained on an annotated corpus. In this paper, we focus on identifying the relevant sentences in the news article from which the highlights were generated. The system we have implemented is named AURUM: AUtomatic Retrieval of Unique information with Machine learning. A full system would also contain a sentence compression step (Knight and Marcu, 2000), but since both steps are largely independent of each other, existing sentence compression or simplification techniques can be applied to the sentences identified by our approach.

The remainder of this paper is organized as follows: The next section describes the relevant work done to date in keyfact extraction and automatic summarization. Section 3 lays out our features and explains how they were learned and estimated. Section 4 presents the experimental setup and our results, and Section 5 concludes with a short discussion.

2 Related Work

As mentioned above, the problem of identifying story highlight lies somewhere between keyword extraction and single-document summarization.

The KEA keyphrase extraction system (Witten et al., 1999) mainly relies on purely statistical features such as term frequencies, using the *tf.idf* measure from Information Retrieval,[1] as well as on a term's position in the text. In addition to *tf.idf* scores, Hulth (2004) uses part-of-speech tags and NP chunks and complements this with machine learning; the latter has been used to good results in similar cases (Turney, 2000; Neto et al., 2002). The B&C system (Barker and Cornacchia, 2000), also used linguistic methods to a very limited extent, identifying NP heads.

INFORMATIONFINDER (Krulwich and Burkey, 1996) requires user feedback to train the system, whereby a user notes whether a given document is of interest to them and specifies their own keywords which are then learned by the system.

Over the last few years, numerous single- as well as multi-document summarization approaches have been developed. In this paper we will focus mainly on single-document summarization as it is more relevant to the issue we aim to address and traditionally proves harder to accomplish. A good example of a powerful approach is a method named Maximum Marginal Relevance which extracts a sentence for the summary only if it is different than previously selected ones, thereby striving to reduce redundancy (Carbonell and Goldstein, 1998).

More recently, the work of Svore et al. (2007) is closely related to our approach as it has also exploited the CNN Story Highlights, although their focus was on summarization and using ROUGE as an evaluation and training measure. Their approach also heavily relies on additional data resources, mainly indexed Wikipedia articles and Microsoft Live query logs, which are not readily available.

Linguistic features are today used mostly in summarization systems, and include the standard features sentence length, n-gram frequency, sentence position, proper noun identification, similarity to title, *tf.idf*, and so-called 'bonus'/'stigma' words (Neto et al., 2002; Leite et al., 2007; Pollock and Zamora, 1975; Goldstein et al., 1999). On the other hand, for most of these systems, simple statistical features and *tf.idf* still turn out to be the most important features.

Attempts to integrate discourse models have also been made (Thione et al., 2004), hand in hand with some of Marcu's (1995) earlier work.

[1] $tf(t,d)$ = frequency of term t in document d.
$idf(t,N)$ = inverse frequency of documents d containing term t in corpus N, $log(\frac{|N|}{|d_t|})$

Regarding syntax, it seems to be used mainly in sentence compression or trimming. The algorithm used by Dorr et al. (2003) removes subordinate clauses, to name one example. While our approach does not use syntactical features as such, it is worth noting these possible enhancements.

3 Approach

In this section we describe which features were used and how the data was annotated to facilitate feature extraction and estimation.

3.1 Training Data

In order to determine the features used for predicting which sentences are the sources for story highlights, we gathered statistics from 1,200 CNN newswire articles. An additional 300 articles were set aside to serve as a test set later on. The articles were taken from a wide range of topics: politics, business, sport, health, world affairs, weather, entertainment and technology. Only articles with story highlights were considered.

For each article we extracted a number of n-gram statistics, where $n \in \{1, 2, 3\}$.

n-gram score. We observed the frequency and probability of unigrams, bigrams and trigrams appearing in both the article body and the highlights of a given story. An important phrase (of length $n \leq 3$) in the article would likely be used again in the highlights. These phrases were ranked and scored according to the probability of their appearing in a given text and its highlights.

Trigger phrases. These are phrases which cause adjacent words to appear in the highlights. Over the entire set, such phrases become significant. We specified a limit of 2 words to the left and 4 words to the right of a phrase. For example, the word *according* caused other words in the same sentence to appear in the highlights nearly 25% of the time. Consider the highlight/sentence pair in Table 1:

highlight:	61 percent of those polled now say it was not worth invading Iraq, poll says
Text:	Now, 61 percent of those surveyed say it was not worth invading Iraq, according to the poll.

Table 1: Example highlight with source sentence.

The word *according* receives a score of 3 since {*invading, Iraq, poll*} are all in the highlight. It should be noted that the trigram {*invading Iraq according*} would receive an identical score, since {*not, worth, poll*} are in the highlights as well.

Spawned phrases. Conversely, spawned phrases occur frequently in the highlights and in close proximity to trigger phrases. Continuing the example in Table 1, {*invading, Iraq, poll, not, worth*} are all considered to be spawned phrases.

Of course, simply using the identities of words neglects the issue of lexical paraphrasing, e.g., involving synonyms, which we address to some extent by using WordNet and other features described in this Section. Table 2 gives an example involving paraphrasing.

highlight:	Sources say men were planning to shoot soldiers at Army base
Text:	The federal government has charged five alleged Islamic radicals with plotting to kill U.S. soldiers at Fort Dix in New Jersey.

Table 2: An example of paraphrasing between a highlight and its source sentence.

Other approaches have tried to select linguistic features which could be useful (Chuang and Yang, 2000), but these gather them under one heading rather than treating them as separate features. The identification of common verbs has been used both as a positive (Turney, 2000) and as a negative feature (Goldstein et al., 1999) in some systems, whereas we score such terms according to a scale. Turney also uses a 'final adjective' measure. Use of a thesaurus has also shown to improve results in automatic summarization, even in multi-document environments (McKeown et al., 1999) and other languages such as Portuguese (Leite et al., 2007).

3.2 Feature Selection

By manually inspecting the training data, the linguistic features were selected. AURUM has two types of features: *sentence features*, such as the position of the sentence or the existence of a negation word, receive the same value for the entire sentence. On the other hand, *word features* are evaluated for each of the words in the sentence, normalized over the number of words in the sentence.

Our features resemble those suggested by previous works in keyphrase extraction and automatic summarization, but map more closely to the journalistic characteristics of the corpus, as explained in the following.

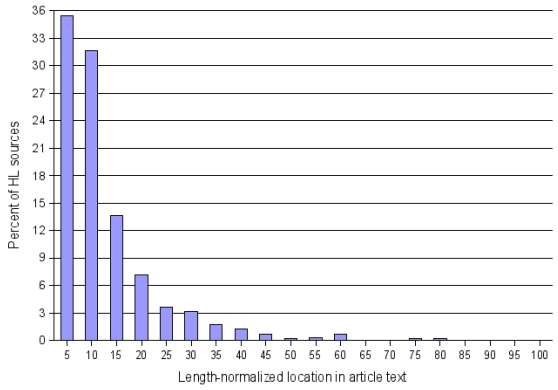


Figure 2: Positions of sentences from which highlights (HLs) were generated.

3.2.1 Sentence Features

These are the features which apply once for each sentence.

Position of the sentence in the text. Intuitively, facts of greater importance will be placed at the beginning of the text, and this is supported by the data, as can be seen in Figure 2. Only half of the highlights stem from sentences in the first fifth of the article. Nevertheless, selecting sentences from only the first few lines is not a sure-fire approach. Table 3 presents an article in which none of the first four sentences were in the highlights. While the baseline found no sentences, AURUM's performance was better.

The sentence positions score is defined as $p_i = 1 - (\log i / \log N)$, where i is the position of the sentence in the article and N the total number of sentences in the article.

Numbers or dates. This is especially evident in news reports mentioning figures of casualties, opinion poll results, or financial news.

Source attribution. Phrasings such as *according to a source* or *officials say*.

Negations. Negations are often used for introducing new or contradictory information: "*Kelly is due in a Chicago courtroom Friday for yet another status hearing,* ***but*** *there's still no trial date in sight.*[2]" We selected a number of typical negation phrases to this end.

Causal adverbs. Manually compiled list of phrases, including *in order to, hoping for* and *because*.

[2]This sentence was included in the highlights

Temporal adverbs. Manually compiled list of phrases, such as *after less than, for two weeks* and *Thursday*.

Mention of the news agency's name. Journalistic scoops and other exclusive nuggets of information often recall the agency's name, especially when there is an element of self-advertisement involved, as in "*...The debates are being held by CNN, WMUR and the New Hampshire Union Leader.*" It is interesting to note that an opposite approach has previously been taken (Goldstein et al., 1999), albeit involving a different corpus.

Story Highlights:

- Memorial Day marked by parades, cookouts, ceremonies
- AAA: 38 million Americans expected to travel at least 50 miles during weekend
- President Bush gives speech at Arlington National Cemetery
- Gulf Coast once again packed with people celebrating holiday weekend

First sentences of article:

1. Veterans and active soldiers unfurled a 90-by-100-foot U. S. flag as the nation's top commander in the Middle East spoke to a Memorial Day crowd gathered in Central Park on Monday.
2. Navy Adm. William Fallon, commander of U. S. Central Command, said America should remember those whom the holiday honors.
3. "Their sacrifice has enabled us to enjoy the things that we, I think in many cases, take for granted," Fallon said.
4. Across the nation, flags snapped in the wind over decorated gravestones as relatives and friends paid tribute to their fallen soldiers.

Sentences the Highlights were derived from:

5. Millions more kicked off summer with trips to beaches or their backyard grills.
6. **AAA estimated 38 million Americans would travel 50 miles or more during the weekend – up 1.7 percent from last year – even with gas averaging $3.20 a gallon for self-service regular**.
7. In the nation's capital, thousands of motorcycles driven by military veterans and their loved ones roared through Washington to the Vietnam Veterans Memorial.
9. President Bush spoke at nearby Arlington National Cemetery, honoring U. S. troops who have fought and died for freedom and expressing his resolve to succeed in the war in Iraq.
21. Elsewhere, Alabama's Gulf Coast was once again packed with holiday-goers after the damage from hurricanes Ivan and Katrina in 2004 and 2005 kept the tourists away.

Table 3: Sentence selection outside the first four sentences (correctly identified sentence by AURUM in **boldface**).

3.2.2 Word Features

These features are tested on each word in the sentence.

'Bonus' words. A list of phrases similar to *sensational, badly, ironically, historic*, identified from the training data. This is akin to 'bonus'/'stigma' words (Neto et al., 2002; Leite et al., 2007; Pollock and Zamora, 1975; Goldstein et al., 1999).

Verb classes. After exploring the training data we manually compiled two classes of verbs, each containing 15-20 inflected and uninflected lexemes, `talkVerbs` and `actionVerbs`. `talkVerbs` include verbs such as {*report, mention, accuse*} and `actionVerbs` refer to verbs such as {*provoke, spend, use*}. Both lists also contain the WordNet synonyms of each word in the list (Fellbaum, 1998).

Proper nouns. Proper nouns and other parts of speech were identified running Charniak's parser (Charniak, 2000) on the news articles.

3.2.3 Sentence Scoring

The overall score of a sentence is computed as the weighted linear combination of the sentence and word scores. The score $\sigma(s)$ of sentence s is defined as follows:

$$\sigma(s) = w_{pos} p_{pos(s)} + \sum_{k=1}^{n} w_k f_k + \sum_{j=1}^{|s|} \sum_{k=1}^{m} w_k g_{jk}$$

Each of the sentences s in the article was tested against the position feature $p_{pos(s)}$ and against each of the sentence features f_k, see Section 3.2.1, where $pos(s)$ returns the position of sentence s. Each word j of sentence s is tested against all applicable word features g_{jk}, see Section 3.2.2. A weight (w_{pos} and w_k) is associated with each feature. How to estimate the weights is discussed next.

3.3 Parameter Estimation

There are various optimization methods that allow one to estimate the weights of features, including generalized iterative scaling and quasi-Newton methods (Malouf, 2002). We opted for generalized iterative scaling as it is commonly used for other NLP tasks and off-the-shelf implementations exist. Here we used YASMET.[3]

We used a development set of 240 news articles to train YASMET. As YASMET is a supervised optimizer, we had to generate annotated data on which it was to be trained. For each document in the development set, we labeled each sentence as to whether a story highlight was generated from it. For instance, in the article presented in Figure 3, sentences 5, 6, 7, 9 and 21 were marked as highlight sources, whereas all other sentences in the document were not.[4]

When annotating, all sentences that were directly relevant to the highlights were marked, with preference given to those appearing earlier in the story or containing more precise information. At this point it is worth noting that while the overlap between different editors is unknown, the highlights were originally written by a number of different people, ensuring enough variation in the data and helping to avoid over-fitting to a specific editor.

4 Experiments and Results

The CNN corpus was divided into a training set and a development and test set. As we had only 300 manually annotated news articles and we wanted to maximize the number of documents usable for parameter estimation, we applied cross-folding, which is commonly used for situations with limited data. The dev/test set was randomly partitioned into five folds. Four of the five folds were used as development data (i.e. for parameter estimation with YASMET), while the remaining fold was used for testing. The procedure was repeated five times, each time with four folds used for development and a separate one for testing. Cross-folding is safe to use as long as there are no dependencies between the folds, which is safe to assume here.

Some statistics on our training and development/test data can be found in Table 4.

Corpus subset	`Dev/Test`	`Train`
Documents	300	1220
Avg. sentences per article	33.26	31.02
Avg. sentence length	20.62	20.50
Avg. number of highlights	3.71	3.67
Avg. number of highlight sources	4.32	-
Avg. highlight length in words	10.26	10.28

Table 4: Characteristics of the evaluation corpus.

[3] A maximum entropy toolkit by Franz Josef Och, `http://www.fjoch.com/YASMET.html`

[4] The annotated data set is available at: `http://www.science.uva.nl/~christof/data/hl/`.

Most summarization evaluation campaigns, such as NIST's Document Understanding Conferences (DUC), impose a maximum length on summaries (e.g., 75 characters for the headline generation task or 100 words for the summarization task). When identifying sentences from which story highlights are generated, the situation is slightly different, as the number of story highlights is not fixed. On the other hand, most stories have between three and four highlights, and on average between four and five sentences per story from which the highlights were generated. This variation led to us to carry out two sets of experiments: In the first experiment (`fixed`), the number of highlight sources is fixed and our system always returns exactly four highlight sources. In the second experiment (`thresh`), our system can return between three and six highlight sources, depending on whether a sentence score passes a given threshold. The threshold θ was used to allocate sentences s_i of article a to the highlight list HL by first finding the highest-scoring sentence for that article $\sigma(s_h)$. The threshold score was thus $\theta * \sigma(s_h)$ and sentences were judged accordingly. The algorithm used is given in Figure 3.

```
initialize HL, s_h
sort s_i in s by σ(s_i)
set s_h = s_0
for each sentence s_i in article a:
    if |HL| < 3
        include s_i
    else if (θ * σ(s_h) ≤ σ(s_i)) && (|HL| ≤ 5)
        include s_i
    else
        discard s_i
return HL
```

Figure 3: Procedure for selecting highlight sources.

All scores were compared to a baseline, which simply returns the first n sentences of a news article. $n = 4$ in the `fixed` experiment. For the `thresh` experiment, the baseline always selected the same number of sentences as `AURUM-thresh`, but from the beginning of the article. Although this is a very simple baseline, it is worth reiterating that it is also a very competitive baseline, which most single-document summarization systems fail to beat due to the nature of news articles.

Since we are mainly interested in determining to what extent our system is able to correctly identify the highlight sources, we chose precision and recall as evaluation metrics. Precision is the percentage of all returned highlight sources which are correct:

$$\text{Precision} = \frac{|R \cap T|}{|R|}$$

where R is the set of returned highlight sources and T is the set of manually identified true sources in the test set. Recall is defined as the percentage of all true highlight sources that have been correctly identified by the system:

$$\text{Recall} = \frac{|R \cap T|}{|T|}$$

Precision and recall can be combined by using the F-measure, which is the harmonic mean of the two:

$$\text{F-measure} = \frac{2(precision * recall)}{precision + recall}$$

Table 5 shows the results for both experiments (`fixed` and `thresh`) as an average over the folds. To determine whether the observed differences between two approaches are statistically significant and not just caused by chance, we applied statistical significance testing. As we did not want to make the assumption that the score differences are normally distributed, we used the bootstrap method, a powerful non-parametric inference test (Efron, 1979). Improvements at a confidence level of more than 95% are marked with "*".

We can see that our approach consistently outperforms the baseline, and most of the improvements—in particular the F-measure scores—are statistically significant at the 0.95 level. As to be expected, `AURUM-fixed` achieves higher precision gains, while `AURUM-thresh` achieves higher recall gains. In addition, for 83.3 percent of the documents, our system's F-measure score is higher than or equal to that of the baseline.

Figure 4 shows how far down in the documents our system was able to correctly identify highlight sources. Although the distribution is still heavily skewed towards extracting sentences from the beginning of the document, it is so to a lesser extent than just using positional information as a prior; see Figure 2.

In a third set of experiments we measured the n-gram overlap between the sentences we have identified as highlight sources and the actual story highlights in the ground truth. To this end we use

System	Recall	Precision	F-Measure	Extracted
Baseline-fixed	40.69	44.14	42.35	240
AURUM-fixed	41.88 (+2.96%*)	**45.40** (+2.85%)	43.57 (+2.88%*)	240
Baseline-thresh	42.91	41.82	42.36	269
AURUM-thresh	**44.49** (**+3.73%***)	43.30 (**+3.53%**)	**43.88** (**+3.59%***)	269

Table 5: Evaluation scores for the four extraction systems.

System	ROUGE-1	ROUGE-2
Baseline-fixed	47.73	15.98
AURUM-fixed	49.20 (**+3.09%***)	16.53 (**+3.63%***)
Baseline-thresh	55.11	19.31
AURUM-thresh	**56.73** (+2.96%*)	**19.66** (+1.87%)

Table 6: ROUGE scores for AURUM-fixed, returning 4 sentences, and AURUM-thresh, returning between 3 and 6 sentences.

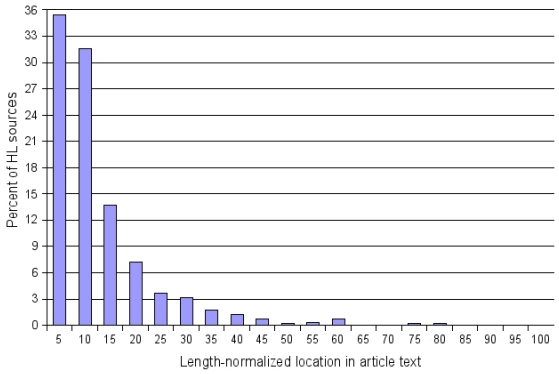


Figure 4: Position of correctly extracted sources by AURUM-thresh.

ROUGE (Lin, 2004), a recall-oriented evaluation package for automatic summarization. ROUGE operates essentially by comparing n-gram co-occurrences between a *candidate* summary and a number of *reference* summaries, and comparing that number in turn to the total number of n-grams in the reference summaries:
ROUGE-n =

$$\frac{\sum_{S \in References} \sum_{ngram_n \in S} Match(ngram_n)}{\sum_{S \in References} \sum_{ngram_n \in S} Count(ngram_n)}$$

Where n is the length of the n-gram, with lengths of 1 and 2 words most commonly used in current evaluations. ROUGE has become the standard tool for evaluating automatic summaries, though it is not the optimal system for this experiment. This is due to the fact that it is geared towards a different task—as ours is not automatic summarization per se—and that ROUGE works best judging between a number of candidate and model summaries. The ROUGE scores are shown in Table 6.

Similar to the precision and recall scores, our approach consistently outperforms the baseline, with all but one difference being statistically significant. Furthermore, in 76.2 percent of the documents, our system's ROUGE-1 score is higher than or equal to that of the baseline, and likewise for 85.2 percent of ROUGE-2 scores. Our ROUGE scores and their improvements over the baseline are comparable to the results of Svore et al. (2007), who optimized their approach towards ROUGE and gained significant improvements from using third-party data resources, both of which our approach does not require.[5]

Table 7 shows the unique sentences extracted by every system, which are the number of sentences one system extracted correctly while the other did not; this is thus an intuitive measure of how much two systems differ. Essentially, a system could simply pick the first two sentences of each article and might thus achieve higher precision scores, since it is less likely to return 'wrong' sentences. However, if the scores are similar but there is a difference in the number of unique sentences extracted, this means a system has gone beyond the first 4 sentences and extracted others from deeper down inside the text.

To get a better understanding of the importance of the individual features we examined the weights as determined by YASMET. Table 8 contains example output from the development sets, with feature selection determined implicitly by the weights the MaxEnt model assigns, where non-discriminative features receive a low weight.

[5]Since the test data of (Svore et al., 2007) is not publicly available we were unable to carry out a more detailed comparison.

Clearly, sentence position is of highest importance, while trigram 'trigger' phrases were quite important as well. Simple bigrams continued to be a good indicator of data value, as is often the case. Proper nouns proved to be a valuable pointer to new information, but mention of the news agency's name had less of an impact than originally thought. Other particularly significant features included temporal adjectives, superlatives and all n-gram measures.

System	Unique highlight sources	Baseline
AURUM-fixed	11.8	7.2
AURUM-thresh	**14.2**	7.6

Table 7: Unique recall scores for the systems.

Feature	Weight	Feature	Weight
Sentence pos.	10.23	Superlative	4.15
Proper noun	5.18	Temporal adj.	1.75
Trigger 3-gram	3.70	1-gram score	2.74
Spawn 2-gram	3.73	3-gram score	3.75
CNN mention	1.30	Trigger 2-gram	3.74

Table 8: Typical weights learned from the data.

5 Conclusions

A system for extracting essential facts from a news article has been outlined here. Finding the data nuggets deeper down is a cross between keyphrase extraction and automatic summarization, a task which requires more elaborate features and parameters.

Our approach emphasizes a wide variety of features, including many linguistic features. These features range from the standard (n-gram frequency), through the essential (sentence position), to the semantic (spawned phrases, verb classes and types of adverbs).

Our experimental results show that a combination of statistical and linguistic features can lead to competitive performance. Our approach not only outperformed a notoriously difficult baseline but also achieved similar performance to the approach of (Svore et al., 2007), without requiring their third-party data resources.

On top of the statistically significant improvements of our approach over the baseline, we see value in the fact that it does not settle for sentences from the beginning of the articles.

Most single-document automatic summarization systems use other features, ranging from discourse structure to lexical chains. Considering Marcu's conclusion (2003) that different approaches should be combined in order to create a good summarization system (aided by machine learning), there seems to be room yet to use basic linguistic cues. Seeing as how our linguistic features—which are predominantly semantic—aid in this task, it is quite possible that further integration will aid in both automatic summarization and keyphrase extraction tasks.

References

Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Conference of the CSCSI, AI 2000*, volume 1882 of *Lecture Notes in Artificial Intelligence*, pages 40–52.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR 1998*, pages 335–336.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics*, pages 132–139.

Wesley T. Chuang and Jihoon Yang. 2000. Extracting sentence segments for text summarization: A machine learning approach. In *Proceedings of the 23rd ACM SIGIR*, pages 152–159.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge Trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 Summarization Workshop*, pages 1–8.

Brad Efron. 1979. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR on Research and Development in IR*, pages 121–128.

Anette Hulth. 2004. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction*. Ph.D. thesis, Department of Computer and Systems Sciences, Stockholm University.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization—step one: Sentence compression. In *Proceedings of AAAI 2000*, pages 703–710.

Bruce Krulwich and Chad Burkey. 1996. Learning user information interests through the extraction of semantically significant phrases. In M. Hearst and H. Hirsh, editors, *AAAI 1996 Spring Symposium on Machine Learning in Information Access*.

Daniel S. Leite, Lucia H.M. Rino, Thiago A.S. Pardo, and Maria das Graças V. Nunes. 2007. Extractive automatic summarization: Does more linguistic knowledge make a difference? In *TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, pages 17–24, Rochester, New York, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55.

Daniel Marcu. 1995. Discourse trees are good indicators of importance in text. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 123–136, Cambridge, MA. MIT Press.

Daniel Marcu. 2003. Automatic abstracting. In *Encyclopedia of Library and Information Science*, pages 245–256.

Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceeding of the 16th national conference of the American Association for Artificial Intelligence (AAAI-1999)*, pages 453–460.

Ani Nenkova. 2005. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *20th National Conference on Artificial Intelligence (AAAI 2005)*.

J. Larocca Neto, A.A. Freitas, and C.A.A Kaestner. 2002. Automatic text summarization using a machine learning approach. In *XVI Brazilian Symp. on Artificial Intelligence*, volume 2057 of *Lecture Notes in Artificial Intelligence*, pages 205–215.

J. J. Pollock and Antonio Zamora. 1975. Automatic abstracting research at chemical abstracts service. *Journal of Chemical Information and Computer Sciences*, 15(4).

Krysta M. Svore, Lucy Vanderwende, and Christopher J.C. Burges. 2007. Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 448–457.

Gian Lorenzo Thione, Martin van den Berg, Livia Polanyi, and Chris Culy. 2004. Hybrid text summarization: Combining external relevance measures with structural analysis. In *Proceedings of the ACL-04*, pages 51–55.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of the ACM Conference on Digital Libraries (DL-99)*.