Abstractive Headline Generation using WIDL-expressions

Radu Soricut and Daniel Marcu
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{radu,marcu}@isi.edu

We present a new paradigm for the automatic creation of document headlines, based on direct transformation of relevant textual information into well-formed textual output. Starting from an input document, we take into account particular properties regarding content selection to automatically create compact representations of weighted finite sets of strings, called WIDL-expressions. A generic natural language generation engine performs the headline generation task, driven by both statistical knowledge encapsulated in WIDL-expressions (representing biases induced by the input document) and statistical knowledge encapsulated in language models (representing biases induced by the target language). Our evaluation shows similar performance in quality with a state-of-the-art, extractive headline generation system, and significant improvements in quality over previously proposed solutions to abstractive headline generation.

1 Introduction

Automatic headline generation has been one of the summarization tasks regarded with most interest in the infancy of the summarization enterprise. The solutions shown to date to best address this task, however, rendered the problem less interesting, as these solutions focused on an extractive approach, in which a relevant sentence is extracted from the input document and compressed to fit length requirements (Dorr et al., 2003; Zajic et al., 2004). This approach exploits the fact that the extracted sentence is correct and often very informative, especially in the news genre, but the resulting headline is often only loosely related to all the relevant terms, and may contain extraneous information. Moreover, such solutions offers little insight on how headline generation is done by humans, and does not offer the opportunity to incrementally improve our scientific understanding of

the intricate mechanisms that direct language generation.

In this article, we present a new paradigm for the automatic creation of document headlines, based on direct transformation of relevant textual information into well-formed textual output. Consider, for instance, a user who wants to be informed, in only one short sentence, of the content of a document, either as a generic summary or as an answer to some specific question. State-of-the-art text analyzers are capable of identifying accurately significant words in the input text (Zhou and Hovy, 2003; Zajic et al., 2004). For example, after processing a document that presents the latest developments of floods in China, such an analyzer builds a list of relevant terms containing Wuhan, Yangtze, chinese, river, rain, and reaches. These relevant words are combined and expanded to create informative phrases, such as Yangtze river valley and rain front. The approach to headline generation we advocate in this article builds sentences starting from these important, individual words and phrases, and uses a generic natural language generation engine to drive the generation process. A possible output created in this fashion reads "RAIN fRONT ON YANGTZE RIVER VALLEY REACHES WUHAN".

This approach to headline generation is summarized in Figure 1. Starting from an input document, a headline-specific module automatically creates compact representations of weighted finite sets of strings, called WIDL-expressions (Soricut, 2006). The weighting scheme used implements specific properties regarding content selection. A generic natural language generation engine, based on WIDL-expressions, accomplishes the generation task. The generation engine is driven by both statistical knowledge encapsulated in WIDL-expressions (representing biases induced by the input document) and statistical knowledge encapsulated in language models (representing biases induced by the target language, English in our case). Our evaluation shows similar performance in quality with a state-of-the-art, extractive headline generation system (Zajic et al., 2004), and significant improvements in quality over a previously proposed solution to abstractive headline

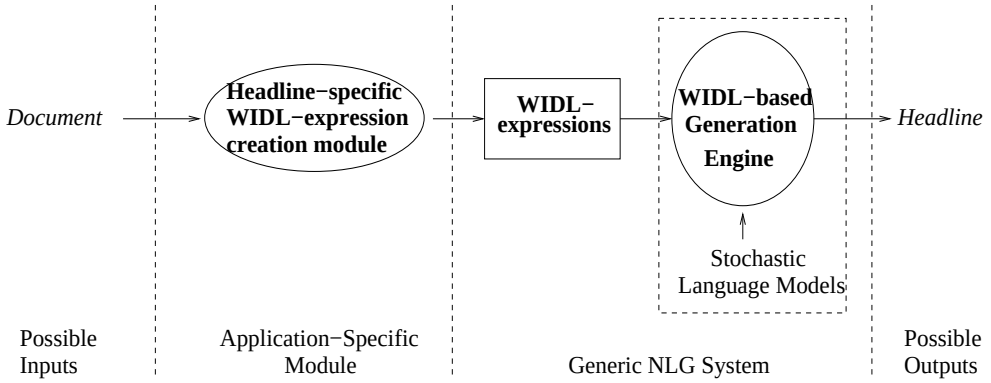

Figure 1
An input document is abstracted by an algorithm into an intermediate representation, called WIDL-expression. A generic NLG engine, which operates on WIDL-expressions, is used as a back-end module for headline generation.

generation (Zhou and Hovy, 2003).

This article is structured as follows. In Section 2, we discuss previous work relevant to headline generation. In Section 3, we provide an introduction to the WIDL-expression formalism, and in Section 4 we describe the algorithm that automatically creates WIDL-expressions starting from an input document. We perform a thorough evaluation of our approach in Section 5, and provide conclusions for our work in Section 6.

2 Previous Approaches to Headline Generation

Two main approaches have been proposed to date for solving the headline generation task. They can be characterized as either extractive, top-down approaches, or abstractive, bottom-up approaches to headline generation.

2.1 Extractive Approaches

Headline generation systems that use the extractive approach generally implement the following steps. First, the most informative sentence in the given document is identified. For the news genre, in particular, the first sentence of the document is usually the most informative. Second, some form of sentence compression is performed, such that the

headline meets some length requirement, usually set to 10 words. Known compression algorithms use either symbolic, rules-based approaches (Chandrasekar et al., 1996; Jing and McKeown, 1999; Canning et al., 2000; Dorr et al., 2003), or stochastic, supervised approaches (Knight and Marcu, 2002).

A symbolic approach to compression, operating on the syntactic structure of the sentence (Dorr et al., 2003), usually produces correct and grammatical sentences, but tends to shrink the sentence too much, and in the process loses important content information. To remedy this problem, another step is added, in which the compressed sentence is "padded" with important content words or phrases. Finding this content information can be achieved by computing noun phrase coreference chains (Bergler et al., 2003) across the input document, or by discovering "topics" using Unsupervised Topic Discovery (Schwartz, 2001).

In one specific proposal (Zajic et al., 2004), an algorithm that identifies "topic keywords" is run over the input document. The retrieved keywords are generally indicative of the topic of the document, and carry important content information. Therefore, the information content of the output headline is increased by adding to the compressed sentence the topic keywords that are not already present. One should note that this system, called Topiary, has provided the best performance at the last headline generation competition, which was run by the National Institute of Standards and Technology (NIST) in 2004.

We call these approaches extractive, because they build their output around a sentence extracted from the input document. Their advantage is that the fluency of the output is ensured by the fact that the extracted sentence is produced by a human (the author of the input document). The disadvantage is that the coverage of their output may be limited. Also, the extractive approach might not prove suitable for building headlines outside the news genre, where the journalistic convention of rendering the most important

information first does not apply.

2.2 Abstractive Approaches

In contrast, abstractive approaches create headlines in a bottom-up manner, starting from important, individual words and phrases, that are glued together to create a fluent sentence. Inspired from work in Machine Translation (Brown et al., 1993), a system developed by Banko et al. (2000) generates headlines using statistical models for content selection and sentence realization. Another abstractive headline generation system, created by Zhou and Hovy (2003), runs first an algorithm that identifies a list of "keywords". The retrieved keywords are then used to extract phrases from the document, which are linked together to create headlines. The advantage of this approach is that these keywords carry a lot of the content information of the document, and therefore a headline that contains many such keywords is likely to be informative. The disadvantage is that it is difficult to create fluent outputs in this manner.

A more recent abstractive headline generation system, proposed by Wan et al. (2005), starts from dependency structures extracted from an input document, and glues them together using n-grams to smooth the transitions between adjacent dependency structures. More general fusion techniques, such as Information Fusion (Barzilay, 2003), are also suitable for abstractive headline generation.

The approach to headline generation described in this article combines some of the ideas of the previous works described. It uses a list of "keywords" to identify the words that carry most of the content in an input document. It also uses syntactic information, extracted from parse trees projected over the sentences of the input document, to build syntactically-driven phrases around the extracted keywords. In addition, it employs a powerful probabilistic framework, called WIDL-expressions, to encode probabilistically content biases present in the input document. The WIDL-framework also allows us to

employ a generic natural language generation system in order to create headlines that are both fluent and informative.

3 Generic Natural Language Generation using WIDL-expressions

3.1 WIDL-expressions

In this section, we briefly present WIDL-expressions, a formal language used to compactly represent probability distributions over finite sets of strings. A thorough presentation of the WIDL formalism can be found in (Soricut, 2006).

Given a finite alphabet of symbols Σ, atomic WIDL-expressions are of the form a, with $a \in \Sigma$. For a WIDL-expression $\omega = a$, its semantics is a probability distribution $\sigma_{\text{widl}}(\omega) : \mathsf{Dom}_\omega \rightarrow [0,1]$, where $\mathsf{Dom}_\omega = \{a\}$ and $\sigma_{\text{widl}}(\omega)(a) = 1$. Complex WIDL-expressions are created from other WIDL-expressions, by employing the following four operators, as well as operator distribution functions δ_i from an alphabet Δ.

Weighted Disjunction. If $\omega_1, \ldots, \omega_n$ are WIDL-expressions, then $\omega = \vee_{\delta_0}(\omega_1, \ldots, \omega_n)$, with $\delta_0 : \{1, \ldots, n\} \rightarrow [0,1]$, specified such that $\Sigma_{k \in \mathsf{dom}(\delta_0)} \delta_0(k) = 1$, is a WIDL-expression. Its semantics is a probability distribution $\sigma_{\text{widl}}(\omega) : \mathsf{Dom}_\omega \rightarrow [0,1]$, where $\mathsf{Dom}_\omega = \bigcup_{i=1}^{n} \mathsf{Dom}_{\omega_i}$, and the probability values are induced by δ_0 and $\sigma_{\text{widl}}(\omega_i)$, $1 \leq i \leq n$. For example, if $\omega = \vee_{\delta_0}(\mathrm{a}, \mathrm{b})$, $\delta_0 = \{1 \rightarrow 0.8,\ 2 \rightarrow 0.2\}$, its semantics is a probability distribution $\sigma_{\text{widl}}(\omega)$ over $\mathsf{Dom}_\omega = \{\mathrm{a}, \mathrm{b}\}$, defined by $\sigma_{\text{widl}}(\omega)(\mathrm{a}) = \delta_0(1) = 0.8$ and $\sigma_{\text{widl}}(\omega)(\mathrm{b}) = \delta_0(2) = 0.2$.

Precedence. If ω_1, ω_2 are WIDL-expressions, then $\omega = \omega_1 \cdot \omega_2$ is a WIDL-expression. Its semantics is a probability distribution $\sigma_{\text{widl}}(\omega) : \mathsf{Dom}_\omega \rightarrow [0,1]$, where Dom_ω is the set of all strings that obey the precedence imposed over the arguments, and the probability values are induced by $\sigma_{\text{widl}}(\omega_1)$ and $\sigma_{\text{widl}}(\omega_2)$. For example, if $\omega_1 = \vee_{\delta_1}(\mathrm{a}, \mathrm{b})$, $\delta_1 = \{1 \rightarrow 0.8,\ 2 \rightarrow 0.2\}$, and $\omega_2 = \vee_{\delta_2}(\mathrm{c}, \mathrm{d})$, $\delta_2 = \{1 \rightarrow 0.6,\ 2 \rightarrow 0.4\}$, then $\omega =$

$\|_{\delta_1}(\times(\text{turkish} \cdot \text{government}), \vee_{\delta_2}(\times(\text{rebels} \cdot \text{fighting}), \times(\text{attacked} \cdot \text{rebels})), \text{in} \cdot \text{iraq}),$
$\delta_1 = \{2\,1\,3 \rightarrow 0.2,\ \text{other perms} \overset{\text{uniform}}{\rightarrow} 0.7, \text{shuffles} \overset{\text{uniform}}{\rightarrow} 0.1\},\ \delta_2 = \{1 \rightarrow 0.65,\ 2 \rightarrow 0.35\}$

Figure 2
An example of a WIDL-expression. The $\times$ operator is used to represent phrasal constructs, while the $\vee$ operator is used to represent weighted choices. The $\|$ operator is used to represent all the possible interleavings of the strings from its arguments, weighted according to the corresponding operator distribution function.

$\omega_1 \cdot \omega_2$ represents a probability distribution $\sigma_{\text{widl}}(\omega)$ over the set $\text{Dom}_\omega = \{\text{ac}, \text{ad}, \text{bc}, \text{bd}\}$, defined by $\sigma_{\text{widl}}(\omega)(\text{ac}) = \delta_1(1)\delta_2(1) = 0.48$, $\sigma_{\text{widl}}(\omega)(\text{ad}) = \delta_1(1)\delta_2(2) = 0.32$, etc.

Weighted Interleave. If $\omega_1, \ldots, \omega_n$ are WIDL-expressions, then $\omega = \|_{\delta_0}(\omega_1, \omega_2, \ldots, \omega_n)$, with $\delta_0 : S \cup \{\text{other perms}\} \cup \{\text{shuffles}\} \rightarrow [0, 1]$, $S \subseteq \text{Perm}_n$, specified such that $\Sigma_{a \in \text{dom}(\delta_0)}\delta_0(a) = 1$, is a WIDL-expression. Its semantics is a probability distribution $\sigma_{\text{widl}}(\omega) : \text{Dom}_\omega \rightarrow [0, 1]$, where Dom_ω consists of all the possible interleavings of strings from Dom_{ω_i}, $1 \leq i \leq n$, and the probability values are induced by δ_0 and $\sigma_{\text{widl}}(\omega_i)$. The distribution function δ_0 is defined either explicitly, over $S \subseteq \text{Perm}_n$ (the set of all permutations of n elements), or implicitly, as $\delta_0(\text{other perms})$. Because the set of argument permutations is a subset of all possible interleavings, δ_0 also needs to specify the probability mass for the strings that are not argument permutations, $\delta_0(\text{shuffles})$. For example, if $\omega = \|_{\delta_0}(\text{a} \cdot \text{b}, \text{c})$, $\delta_0 = \{1\ 2 \rightarrow 0.80,\ \text{other perms} \overset{\text{uniform}}{\rightarrow} 0.15, \text{shuffles} \overset{\text{uniform}}{\rightarrow} 0.05\}$, its semantics is a probability distribution $\sigma_{\text{widl}}(\omega)$, with domain $\text{Dom}_\omega = \{\text{abc}, \text{cab}, \text{acb}\}$, defined by $\sigma_{\text{widl}}(\omega)(\text{abc}) = \delta_0(1\ 2) = 0.80$, $\sigma_{\text{widl}}(\omega)(\text{cab}) = \frac{\delta_0(\text{other perms})}{1} = 0.15$, $\sigma_{\text{widl}}(\omega)(\text{acb}) = \frac{\delta_0(\text{shuffles})}{1} = 0.05$.

Lock. If ω' is a WIDL-expression, then $\omega = \times(\omega')$ is a WIDL-expression. The semantic mapping $\sigma_{\text{widl}}(\omega)$ is the same as $\sigma_{\text{widl}}(\omega')$, except that Dom_ω contains strings in which no additional symbol can be interleaved. For example, if $\omega = \|_{\delta_0}(\times(\text{a} \cdot \text{b}), \text{c})$, $\delta_0 = \{1\ 2 \rightarrow 0.80,\ \text{other perms} \rightarrow 0.20\}$, its semantics is a probability distribution $\sigma_{\text{widl}}(\omega)$, with domain $\text{Dom}_\omega = \{\text{cab}, \text{abc}\}$, defined by $\sigma_{\text{widl}}(\omega)(\text{abc}) = \delta_0(1\ 2) = 0.80$, $\sigma_{\text{widl}}(\omega)(\text{cab}) = \frac{\delta_0(\text{other perms})}{1} = 0.20$.

In Figure 2, we show a more complex WIDL-expression. The probability distribution δ_1 associated with the operator $\|_{\delta_1}$ assigns probability 0.2 to the argument order 2 1 3; from a probability mass of 0.7, it assigns uniformly, for each of the remaining $3! - 1 = 5$ argument permutations, a permutation probability value of $\frac{0.7}{5} = 0.14$. The remaining probability mass of 0.1 is left for the 12 shuffles associated with the unlocked expression $\mathsf{in} \cdot \mathsf{iraq}$, for a shuffle probability of $\frac{0.1}{12} = 0.008$. The list below enumerates some of the $\langle \mathsf{string}, p(\mathsf{string}) \rangle$ pairs that belong to the probability distribution defined by our example:

$$\begin{array}{rl} \mathsf{rebels\ fighting\ turkish\ government\ in\ iraq} & 0.130 = 0.20_{\delta_1} \times 0.65_{\delta_2} \\ \mathsf{in\ iraq\ attacked\ rebels\ turkish\ government} & 0.049 = 0.14_{\delta_1} \times 0.35_{\delta_2} \\ \mathsf{in\ turkish\ government\ iraq\ rebels\ fighting} & 0.005 = 0.008_{\delta_1} \times 0.65_{\delta_2} \end{array}$$

3.2 Intersecting WIDL-expressions and n-gram Language Models in a Log-linear Framework

Let us assume a finite set E of strings over a finite alphabet Σ, representing the set of possible realizations. In a log-linear framework, we have a vector of feature functions $\overline{h} = \langle h_0, h_1, \ldots, h_M \rangle$, and a vector of parameters $\overline{\lambda} = \langle \lambda_0, \lambda_1, \ldots, \lambda_M \rangle$. For any $e \in E$, the interpolated probability $P(e)$ can be written under a log-linear model as in Equation 1:

$$P(e) = \frac{\exp[\Sigma_{m=0}^{M} \lambda_m h_m(e)]}{\Sigma_{e'} \exp[\Sigma_{m=0}^{M} \lambda_m h_m(e')]} \tag{1}$$

We can formulate the search problem of finding the most probable realization e under this model as shown in Equation 2, and therefore we do not need to be concerned about computing expensive normalization factors.

$$\arg\max_e P(e) = \arg\max_e \exp[\Sigma_{m=0}^{M} \lambda_m h_m(e)] \tag{2}$$

For a given WIDL-expression ω over Σ, the set E is defined by $\mathsf{dom}(\sigma_{\mathrm{widl}}(\omega))$, and feature function h_0 is taken to be $\sigma_{\mathrm{widl}}(\omega)$. Any language model we want to employ may be added in Equation 2 as a feature function h_i, $i \geq 1$.

The search problem defined by Equation 2, for a WIDL-expression ω (which provides

feature function h_0) and M n-gram language models (which provide feature functions $h_1, \ldots, h_M$), can be solved by an A*-style search algorithm (Soricut and Marcu, 2006). This algorithm, called WIDL-NGLM-A*, finds a string with maximum probability under the model defined by Equation 1, i.e., an interpolation of the distribution probability given by a WIDL-expression and the M n-gram language model distribution probabilities.

4 Automatic Creation of WIDL-expressions for Headline Generation

In this section, we present an algorithm used to implement a headline-specific WIDL-expression creation module, which automatically creates WIDL-expressions starting from input documents (see Figure 1). The resulting WIDL-expressions are fed to a generic natural generation system to automatically generate headlines.

Starting from an input document, we first run the algorithm presented in (Zhou and Hovy, 2003) to extract a weighted list of keywords from an input document (see the example in Figure 3(a), for which the input is a document describing incidents at the Turkey-Iraq border and the surrounding region). Next, we parse the input document using an in-house implementation of a state-of-the-art statistical parser (Collins, 2003), extract the lexical dependencies for all keywords, and create a list of phrases associated with each keyword. For each such phrase list, we associate a probability distribution computed using the frequency of these phrases (we assume that higher frequency is indicative of increased importance) and the position of these phrases within the document (if the phrase occurs multiple times, the first occurrence is considered to determine position; we assume that proximity to the beginning of the document is also indicative of importance). For the current running example, this list is presented in Figure 3(b).

The algorithm which produces WIDL-expressions combines first the lexical-dependency phrases for each keyword i under a $\vee_{\delta_i}$ operator. Additionally, a special eps expression is also specified as an argument for each $\vee_{\delta_i}$ operator, which accounts for the possibility of

(a) Keywords

iraq 0.32, syria 0.25, rebels 0.22, kurdish 0.17, turkish 0.14, attack 0.10

(b) Lexical-dependency Phrases

iraq	northern iraq 0.48, in iraq 0.39, iraq and iran 0.12
syria	into syria 0.55, and syria 0.45
rebels	attacked rebels 0.63, rebels fighting 0.28, kurdish rebels 0.09
turkish	turkish government 0.77, turkish soldiers 0.23

(c) WIDL-expression

$$
\begin{aligned}
\|_{\delta_0}(\quad & \vee_{\delta_1}(\times(\text{northern} \cdot \text{iraq}),\ \times(\text{in} \cdot \text{iraq}),\ \times(\text{iraq} \cdot \text{and} \cdot \text{iran}),\ \text{eps}\ ), \\
& \vee_{\delta_2}(\times(\text{into} \cdot \text{syria}),\ \times(\text{and} \cdot \text{syria}),\ \text{eps}\ ), \\
& \vee_{\delta_3}(\times(\text{attacked} \cdot \text{rebels}),\ \times(\text{rebels} \cdot \text{fighting}),\ \times(\text{kurdish} \cdot \text{rebels}),\ \text{eps}\ ), \\
& \vee_{\delta_4}(\times(\text{turkish} \cdot \text{government}),\ \times(\text{turkish} \cdot \text{soldiers}),\ \text{eps}\ ) \\
)\ & \\
& \delta_0 = \{\text{perms} \overset{\text{uniform}}{\rightarrow} 1\} \\
& \delta_1 = \{1 \rightarrow 0.48,\ 2 \rightarrow 0.39,\ 3 \rightarrow 0.12,\ 4 \rightarrow 0.01\} \\
& \delta_2 = \{1 \rightarrow 0.51,\ 2 \rightarrow 0.40,\ 3 \rightarrow 0.09\} \\
& \delta_3 = \{1 \rightarrow 0.63,\ 2 \rightarrow 0.28,\ 3 \rightarrow 0.09,\ 4 \rightarrow 0.008\} \\
& \delta_4 = \{1 \rightarrow 0.75,\ 2 \rightarrow 0.21,\ 3 \rightarrow 0.05\}
\end{aligned}
$$

(d) Output headline (most likely realization)

TURKISH GOVERNMENT ATTACKED REBELS IN IRAQ AND SYRIA

Figure 3
From an input document, a weighted list of keywords is extracted(a), from which phrases are proposed (b) and combined in a WIDL-expression (c). The output (d) for our automatic headline generation system is also shown.

dropping keyword i. (The eps expression is treated as a regular Σ-label within the WIDL formalism, but it is filtered out from the output headline.) For a given keyword, the probability values associated with each phrase are used to specify the probability associated with each argument of the δ_i specification, after setting aside a small probability mass for the eps choice. The probability of the eps choice is computed such that its cost (i.e., the $-\log$ of the probability) is twice the cost of the most expensive non-eps choice. All of the $\vee_{\delta_i}$-headed expressions are finally combined into a single WIDL-expression using a $\|_{\delta_0}$ operator, $\delta_0 = \{\mathsf{perms} \overset{\mathsf{uniform}}{\rightarrow} 1\}$ (uniform probability over all the possible orders of the keyword phrases). The WIDL-expression in Figure 3(c) results from applying this algorithm to the phrases in our example. On average, a WIDL-expression created by this algorithm, using $m = 6$ keywords and an average of $k = 4$ lexical-dependency phrases per keyword, compactly encodes a candidate set of about 3 million possible realizations.

Finally, we generate headlines from WIDL-expressions using the WIDL-NGLM-A* algorithm (Soricut and Marcu, 2006), which interpolates the probability distributions represented by the WIDL-expressions with n-gram language model distributions. The output presented in Figure 3(d) is the most likely headline realization produced by this algorithm.

5 Headline Generation Evaluation

To evaluate the accuracy of the WIDL-based headline generation system, we use the documents from the Document Understanding Conference (DUC–2003) evaluation competition. Half of these documents are used as development set (283 documents), on which the $\overline{\lambda}$ weights (Equation 2) are discriminatively trained (Och, 2003) using ROUGE_2 (Lin, 2004) as the objective function. The other half is used as test set (273 documents). We automatically measure performance by comparing the produced headlines against one reference headline produced by a human using ROUGE_2.

5.1 The ROUGE$_n$ Metric

The ROUGE$_n$ is a recall-focused family of metrics that takes as parameter a non-negative integer n, and employs a list of stop-words SW and a word-stemming function ST (Porter, 1980). Given a set of candidate outputs, one uses a set of *References* to compute, for each reference R, the multi-set $S(R,n)$ of n-grams obtained from the reference R after stemming the unigrams using function ST and eliminating the unigrams found in SW. A recall score is computed as:

$$R(n) = \frac{\sum_{R \in References} \sum_{ngram \in S(R,n)} Count_{clip}(ngram)}{\sum_{R \in References} \sum_{ngram \in S(R,n)} Count(ngram)}$$

where $Count(ngram)$ is the number of n-gram counts, and $Count_{clip}(ngram)$ is the maximum number of *ngram* co-occurrences in the reference and its corresponding candidate. Because the denominator in the above formula consists of a sum over the reference answers, $R(n)$ is essentially a recall-oriented measure, which penalizes incomplete candidates. This recall value, however, has the potential to become higher for longer candidates. The usual fix for this problem is to provide a cap for the size of the candidates, such as 10 words for headlines or 250 words for summaries (Lin, 2004).

The approach we take is different, along the lines proposed by Soricut and Brill (2004). We define a wordiness penalty measure, WP, as follows:

$$WP = \begin{cases} 1 & \text{if } \frac{r}{c} \geq 1 \\ e^{1-\frac{c}{r}} & \text{if } \frac{r}{c} < 1 \end{cases}$$

where c equals the sum of the lengths of the candidates, and r equals the sum of the lengths desired for the candidates. For instance, if the target length of a candidate is 10, then r is set to be 10 times the number of candidates in the test set. The ROUGE$_n$ family of metrics can be now defined as follows:

$$\text{ROUGE}_n = WP \cdot \exp[\sum_{k=1}^{n} \log R(k)]$$

This definition of the ROUGE$_n$ metrics allows us to learn, via discriminative training,

λ_{WIDL}	$\lambda_{\text{General English}}$	$\lambda_{\text{Document English}}$	$\lambda_{\text{Word Count}}$	$\lambda_{\text{Phrase Count}}$
0.156187	0.009044	0.291960	-0.460975	-0.081834

Table 1
Weight values assigned by discriminative training to the model features that contribute to determining the most likely headline realization.

to finely compensate between the gain obtained by longer candidates and the wordiness penalty imposed by the metric. The effect is that our model's parameters learn to produce headlines that have 10 words on average, such that the penalty induced by the wordiness measure *WP* is minimal.

5.2 Feature Evaluation

We first perform a series of experiments that validate our choice for the features used by Equation 2. These features are defined as follows. First, the probability distribution associated with each WIDL-expression is used as a feature function. Second, we train a large trigram language model on 170M English words from the Wall Street Journal, using the SRI Language Model Toolkit (Stolcke, 2002), and smoothed using modified Kneser-Ney smoothing (Goodman, 2001). We call this model the General English model, and use it to model fluency.

For each input document, we also train on-the-fly a document-specific trigram language model, also using the SRI Language Model Toolkit and modified Kneser-Ney smoothing. This model, called the Document English model, accounts for both fluency and content validity. We also use a Word Count model (which counts the number of words in a proposed realization) and a Phrase Count model (which counts the number of phrases in a proposed realization), which allow one to learn to produce headlines that have restrictions in the number of words allowed (10, in this case).

In summary, we use Equation 2 with feature function h_0 for the WIDL-expressions, two feature functions h_1 and h_2 for the General English and Document English models,

respectively, and two feature functions h_3 and h_4 for the Word Count and Phrase Count models, respectively. The interpolation weights for Equation 2,

$$\overline{\lambda} = \langle \lambda_{\text{WIDL}}, \lambda_{\text{General English}}, \lambda_{\text{Document English}}, \lambda_{\text{Word Count}}, \lambda_{\text{Phrase Count}} \rangle,$$

are trained using discriminative training (Och, 2003) using ROUGE_2 as the objective function, on the development set. The values obtained for the $\overline{\lambda}$ weights are given in Table 1. These values reflect the fact that the probability distributions associated with the WIDL-expressions, reflecting input-document bias, are important (weight 0.156187). The weight of the General English model is not as high as the weight of the Document English model (0.009044 versus 0.291960), which indicates that the usefulness of a language model for this task is correlated with the fit between the model training data and the test data. The negative weight for the Word Count model (weight -0.460975) is necessary to penalize long realizations, and, together with the rest of the weights, contributes to the log-linear model's ability to prefer realizations that have 10 words on average. Finally, the small negative weight for the Phrase Count model (weight -0.081834) indicates that the model has learnt to prefer fewer phrases, which in the context of a constant word length translates into preferring longer phrases. Longer phrases have the potential to provide better fluency.

To confirm our intuitions regarding the importance of each feature function, together with their associated weight, we perform a series of experiments in which we eliminate, one at a time, the feature functions employed. The one feature we cannot eliminate is the Word Count feature, which is needed for the model to learn to produce headlines of the required length. The decoder used for all these experiments implements the WIDL-NGLM-A* algorithm. The results are presented in Table 2.

We present both average times needed to "decode" a WIDL-expression and headline quality measures. When using all the features, the average time is 8.7 seconds per WIDL-

Features	**Speed (sec./exp.)**	♯(UNI)	♯(BI)	ROUGE_1	$\textbf{ROUGE}_2$
All	8.7	562	126	25.5	12.9
No WIDL weights	24.0	416	110	18.9	10.4
No General English	6.8	557	122	25.3	12.6
No Document English	1.5	478	106	21.7	10.9
No Phrase Model	10.9	549	124	24.9	12.7

Table 2
Feature function evaluation. The importance of each individual feature is measured in terms of the impact on the speed of the decoder, and also in terms of the impact on the headline quality.

expression, on a 3.0GHz CPU machine under a Linux OS. To arrive at the actual average time needed to obtain a headline starting from an input document, one needs to add to this figure the average time needed to parse the input document (both the time to extract the keyword list and the time to build a WIDL-expression given the keywords and the parsed document are under 1 second, and therefore negligible for this estimate). Given that the average number of sentences in our documents is 24, and that the average time to parse one sentence using our parser is 0.8 seconds, we estimate that the average time to arrive at a headline starting from an input document is under 30 seconds (about 20 seconds for parsing, and under 10 seconds for decoding a WIDL-expression).

We measure the quality of the proposed headline using a reference headline produced by the author of the input document. We report four figures: raw numbers for unigram and bigram matches between the proposed headline and the reference headline, and also ROUGE_1 and ROUGE_2 scores. Of these four numbers, the ROUGE_2 score is the one that is known to correlate best with human judgment on summarization evaluation (DUC, 2003–2004). The other three numbers are useful in providing clues on whether the difference in ROUGE_2 scores is due to less content coverage (indicated by a decrease in ♯(UNI) and ROUGE_1 score), decreased fluency (smaller ♯(BI)), or both.

The results in Table 2 confirm our intuition that WIDL-induced biases (via WIDL probability distributions) are important. Without them, the decoder has more difficulties

in finding the best realization (average time 24 seconds versus 8.7 seconds), as the only bias left is the one induced by the language models. Moreover, we see a significant drop in headline quality, as measured by the ROUGE_2 score (10.4 versus 12.9). This drop is due mostly to poorer content coverage (416 versus 562 $\sharp$(UNI)), as language models can only poorly approximate content importance.

When the General English feature is dropped, the drop in ROUGE_2 is small (12.6 versus 12.9), and seems to be due to small drops in both content coverage and fluency. On the other hand, when the Document English feature is dropped, the drop in ROUGE_2 is significant (10.9 versus 12.9), due to drops in both content coverage (478 versus 562 $\sharp$(UNI)) and fluency (106 versus 126 $\sharp$(BI)). However, the time savings obtained in this case (1.5 seconds versus 8.7 seconds) are considerable, and suggest that most of the decoding time for the all-feature model is spent in the I/O operations needed to load document-specific language models for each expression decoded. Finally, without the Phrase Model feature, decoding seems to take longer (10.9 seconds versus 8.7 seconds), and the drop in ROUGE_2 is small (12.7 versus 12.9).

In conclusion, this evaluation shows that the most important features are the input document biases represented by the WIDL-expressions and the language model trained on the input document. The other features seem to have a smaller impact, but nevertheless contribute positively to the overall speed and accuracy of the all-feature model.

5.3 Evaluation against Other Methods

In this section, we evaluate the performance of our WIDL-based headline generation systems against previously-proposed methods for headline generation. We compare the performance of several extractive algorithms against several abstractive algorithms (see Section 2). For the extractive algorithms, **Lead10** is a baseline which simply proposes as headline the first 10 words of the lead sentence. **HedgeTrimmer**† is our implementation

System	Av. Len.	♯(UNI)	♯(BI)	$ROUGE_1$	**$ROUGE_2$**
Extractive					
Lead10	9.9	458	114	20.8	11.1
HedgeTrimmer†	7.4	399	104	18.1	9.9
Topiary‡	9.9	576	115	26.2	12.5
Abstractive					
Keywords	9.9	585	22	26.6	5.5
Webcl	7.3	311	76	14.1	7.5
WIDL	10.0	562	126	25.5	**12.9**

Table 3
Headline generation evaluation. We compare extractive algorithms against abstractive algorithms, including a WIDL-based headline generation algorithm.

of the Hedge Trimer algorithm (Dorr et al., 2003), and **Topiary**‡ is our implementation of the Topiary algorithm (Zajic et al., 2004). For the abstractive algorithms, **Keywords** is a baseline that proposes as headline the sequence of keywords sorted according to their scores (from highest to lowest), and **Webcl** is the actual output of the system described in (Zhou and Hovy, 2003). The WIDL system uses the all-feature model (Section 5.2) and the WIDL-NGLM-A* search algorithm.

The results of this experiment show that the WIDL-based approach to generation is capable of creating headlines that comparable favorably, in both content and fluency, with extractive, state-of-the-art results (Zajic et al., 2004). The best performance is obtained by the WIDL system, 12.9 $ROUGE_2$ score, followed by the system implementing the Topiary solution, 12.5 $ROUGE_2$ score. The Topiary solution gets slightly better coverage (576 versus 562 ♯(UNI)), while the WIDL-based system gets better fluency (126 versus 115 ♯(BI)). The difference between the two $ROUGE_2$ scores is not statistically significant at 95% confidence, but it is statistically significant compared to all the other scores presented in Table 3. It is interesting to notice that the Keywords baseline gets the highest $ROUGE_1$ score (26.6), but the lowest $ROUGE_2$ score (5.5). As the output of the Keywords system is far from what we might consider a proper headline, this result evidences the appropriateness of using $ROUGE_2$ over $ROUGE_1$ for automatic headline evaluation.

One should also note that this evaluation was conducted such that the comparisons are more direct and meaningful, as the compared implementations use the same algorithms for parsing (Collins, 2003) and finding keywords (Zhou and Hovy, 2003).

5.4 Evaluation of Extractiveness versus Abstractiveness

In the previous experiments, we grouped systems into **Extractive** and **Abstractive**, based on the intuition that the extractive systems creates headlines starting from a sentence extracted from the input document, whereas the abstractive systems create their output more freely. The final evaluation we perform tries to quantify the "abstractiveness" quality of the systems considered in our evaluation.

Toward this end, we propose a way of measuring this "abstractiveness" quality. First, given an input document and a proposed hypothesis headline, we determine the document sentence which is the best match for the hypothesis, i.e., has the maximum number of word matchings (case insensitive). For a given test set, we find the best-match sentence for each proposed headline, and compute two statistics: the spread of the best-match, described as the percent figure for having the best-match as the first sentence, second sentence, and third or beyond sentence; and the percent of word matchings between the hypothesis headline and the best-match sentence.

In Table 4, we present the result of this evaluation. First, we consider the reference headlines as our hypothesis headlines, and compute the above statistics. The results show that the human-created headline matches best the first sentence of the document 65.7% of the time, the second sentence 8.1%, and the third sentence or beyond 26.2% of the time. The percent of word matches between the best-match and the reference headline is 57.6%, i.e., only about half of the words in a human-created headline (including non-content words) are to be found in a single sentence in the document.

In contrast, the systems labeled as **Extractive** have very different statistics: they

System	Spread of Best-match			Hyp. coverage
	1st sent.	2nd sent.	≥3rd	in Best-match
Reference				
Human author	65.7%	8.1%	26.2%	57.6%
Extractive				
Lead10	100%	0.0%	0.0%	100%
HedgeTrimmer†	100%	0.0%	0.0%	95.4%
Topiary‡	100%	0.0%	0.0%	95.1%
Abstractive				
Keywords	99.6%	0.4%	0.0%	92.8%
Webcl	59.0%	7.4%	33.6%	60.8%
WIDL	92.2%	2.6%	5.2%	85.5%

Table 4
Evaluation of the "extractiveness" versus "abstractiveness". The statistics computed for the human-authored references are indicative of the "abstractiveness" property.

have the best-match as the first sentence 100% of the time, and the percent of word matches between the best-match and the proposed headline is above 95%, i.e., almost exclusive first-sentence coverage. On the other hand, the systems labeled **Abstractive** have statistics that look closer to the reference headline statistics. The Webcl system creates headlines that are best-matched by the first sentence of the document 59.0% of the time, by the second sentence 7.4%, and by the third sentence or beyond 33.6% of the time. The percent of word matches between the best-match and the Webcl headlines is 60.8%, which makes for statistics that are remarkably closer to those obtained using the human-created headlines. If the human-created headlines are to be considered truly abstractive, then, by this metric, the Webcl is the most "abstractive" system. By the same metric, the WIDL system is characterized by statistics that fall somewhere between the extractive and abstractive limits: its headlines are best-matched by the first sentence of the document 92.2%, by the second sentence 2.6%, and by the third sentence or beyond 5.2% of the time. The percent of word matches between the best-match and its headlines is 85.5%. Although these numbers may appear somewhat closer to the extractive side than the generation paradigm would suggest, the explanation lies in the bias induced by the Keywords algorithm: the extracted keywords have a best-match in the first sentence

Good
SCIENTISTS ARE EXPLORING CHAOS OF BRAIN TO TREAT SCHIZOPHRENIA
WATER IS LINK BETWEEN CLUSTER OF E. COLI CASES
SRI LANKA 'S JOINT VENTURE TO EXPAND EXPORTS
THREE GORGES PROJECT IN CHINA HAS WON APPROVAL
OPPOSITION TO EUROPEAN UNION SINGLE CURRENCY EURO
Almost good
FINAL DECISION ON ISSUE IN INDIA AND BANGLADESH BORDER
THREE GORGES PROJECT SAID THAT IS TO BE COMPLETED BY 2003
OF INDIA AND BANGLADESH WATER BARRAGE
EAST TIMOR 'S THREAT TO LAUNCH ATTACKS FROM INDONESIA
INVESTIGATORS TO PROBE IN KOSOVO WAR CRIMES
Terrible
FAMILY HISTORY OF SCHIZOPHRENIA STUDY WITH RISK OF BIRTH
STATE HISTORY OF E. COLI POISONING WORST ONE CHILD
NORTHERN IRELAND PEACE PROCESS WOUNDED SICK AID WITHOUT BORDERS
ELUSIVE SUBATOMIC PARTICLE PHYSICISTS NEUTRINO MASS IN JAPAN
POLICE HAVE KILLED PEOPLE ARCTIC COLD WAVE THROUGHOUT

Table 5
Headlines generated automatically using a WIDL-based sentence realization system.

99.6% of the time, and there is a 92.8% match between the keywords and the best-match, i.e., the first sentence.

These results seem to suggest that the difficulty in differentiating high quality abstractive methods from extractive methods, using this type of statistics, may lie not in the way the systems are built, but in the genre of the documents used for evaluation. Indeed, the journalistic style in which the news documents used in the DUC evaluations are written allows for little room in differentiating extractive versus abstractive systems at similar levels of quality.

5.5 Examples of Headlines

To conclude this section on evaluation, we present in Table 5 a sample of headlines produced by our WIDL-based headline generation system. We selected by hand five output headlines that we considered indistinguishable from a human-created headline (the **Good** batch of examples). We also selected five output headlines that are close to being human-level, but for which certain words or meanings are not quite correct (the

batch labeled **Almost good**). For instance, the preposition IN in the headline FINAL DECISION ON ISSUE IN INDIA AND BANGLADESH BORDER is not correctly used, and replacing it with REGARDING would make for a perfect headline. Also, the headline EAST TIMOR 'S THREAT TO LAUNCH ATTACKS FROM INDONESIA has a different meaning than the one present in the document because of the incorrect usage of the preposition FROM.

The last batch of example, labeled **Terrible**, provides even more insight into what is not properly working in our current system. From a syntactic perspective, one can see that the different syntactic phrases that were employed could not be linked together properly by the language models. From a semantic perspective, it becomes clear that the individual meanings of the phrases are not enough to ensure an overall meaningful outcome. This seems to suggest that more powerful language models, ones that are capable of properly accounting for language syntax and semantics, could be used by our generation paradigm to provide a more powerful mechanism for controlling language production.

6 Conclusions

In this article, we have presented a headline-generation application based on WIDL-expressions. This application takes as input text documents, and creates, from bits and pieces of textual information, WIDL-expressions that compactly represent many possible headline realizations. The WIDL-expressions are intersected with both general and document-specific language models, and the best scoring realizations are presented as the output headlines.

This abstractive, bottom-up approach is compared and contrasted with extractive approaches, which build their output starting from an existing sentence in the document. The evaluations we carried show that the WIDL-based headline generation system performs at the same level of accuracy as an extractive, state-of-the-art system. At the same time, it outperforms a previously-proposed abstractive approach by a wide margin.

We have also proposed a measure to compare and contrast human-generated headlines against automatically-generated headlines. This measure allows us to quantitatively assess the current mismatch between certain properties of human-generated headlines and headlines generated by either extractive and abstractive means. While extractive approaches have no opportunity to improve with respect to these mismatches, we believe that the abstractive approach advocated in this article has the potential to go a long way towards rendering these mismatches less prominent.

As our examples of WIDL-generated headlines show, this approach has the potential to produce headlines that are indistinguishable from human-generated ones. When this potential goes unfulfilled, we can often pinpoint the cause of failure, not as a failure of the approach as a whole, but as a failure to properly model certain language requirements. This opens up new directions, not only for summarization research, but for language research in general. For instance, syntax-driven stochastic language models (Charniak, 2001) can be integrated in the WIDL-based approach to generation (Soricut, 2006), and have the potential to fix many of the grammaticality problems we observe. By the same token, language models that capture semantic aspects of language are yet to be defined and employed towards fixing problems caused by our semantics-agnostic system. Instead of considering the headline generation problem "solved" simply because extractive solutions happen to perform well for the news genre, we see it as a well-defined opportunity to drive progress for many yet unsolved language engineering problems.

Acknowledgments This work was partially supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

References

Michele Banko, Vibhu Mittal, and Michael Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL–2000)*, pages 318–325, Hong Kong, October 1–8.

Regina Barzilay. 2003. *Information Fusion for Multidocument Summarization: Paraphrasing and Generation.* Ph.D. thesis, Columbia University.

Sabine Bergler, Rene Witte, Michelle Khalife, Zhuoyan Li, and Frank Rudzicz. 2003. Using knowledge-poor coreference resolution for text summarization. In *Proceedings of the Document Understanding Conference (DUC 2003)*, Edmonton, Alberta, Canada, May 31 - June 1.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Y. Canning, J. Tait, J. Archibald, and R. Crawley. 2000. Cohesive generation of syntatically simplified newspaper text. In *Proceedings of the Workshop on Robust Methods in Analysis of Natural Language Data*, pages 145–150.

R. Chandrasekar, Christy Doran, and Srinivas Bangalore. 1996. Motivations and methods for text simplification. In *Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING '96)*, Copenhagen, Denmark.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*.

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

Joshua Goodman. 2001. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research.

Hongyan Jing and Kathleen R. McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the 22nd Conference on Research and Development in Information Retrieval (SIGIR–99)*, Berkeley, CA, August 15–19.

Kevin Knight and Daniel Marcu. 2002. Statistics-based summarization — step one: Sentence compression. *Artificial Intelligence*, 139(1).

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, July 25 - 26.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the ACL*, pages 160–167.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, pages 130–137.

Richard Schwartz. 2001. Unsupervised topic discovery. In *Proceedings of Workshop on Language Modeling and Information Retrieval, pp.72-77*, pages 72–77.

Radu Soricut and Eric Brill. 2004. A unified framework for automatic evaluation using N-gram co-occurrence statistics. In *Proceedings of the Association for Computational Linguistics Conference*, Barcelona, Spain, July 22–25.

Radu Soricut and Daniel Marcu. 2006. Stochastic language generation using widl-expressions and its application in machine translation and summarization. In *Proceedings of the Association for Computational Linguistics Conference*.

Radu Soricut. 2006. *Natural Language Generation for Text-to-Text Applications Using an Information-Slim Representation*. Ph.D. thesis, University of Southern California.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Stephen Wan, Robert Dale, Mark Dras, and Cecile Paris. 2005. Statistically generated summary sentences: A preliminary evaluation using a dependency relation precision metric. In *Proceedings of the Workshop on Using Corpora for Natural Language Generaton*.

David Zajic, Bonnie J. Dorr, and Richard Schwartz. 2004. BBN/UMD at DUC-2004: Topiary. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on Document Understanding*, pages 112–119, Boston, MA.

Liang Zhou and Eduard Hovy. 2003. Headline summarization at ISI. In *Proceedings of the Document Understanding Conference (DUC 2003)*, Edmonton, Alberta, Canada, May 31 - June 1.