Cross-Language Headline Generation for Hindi

BONNIE DORR, DAVID ZAJIC
Institute for Advanced Computer Studies, University of Maryland, College Park
RICHARD SCHWARTZ
BBN Technologies, Columbia, Maryland

This paper presents new approaches to *headline generation* of English newspaper texts, with an eye toward the production of document surrogates for document selection in cross-language information retrieval. This task is difficult because the user must make decisions about relevance based on (often poor) translations of retrieved documents. To facilitate the decision-making process we need translations that can be assessed rapidly and accurately; our approach is to provide an English headline for the non-English document. We describe two approaches to headline generation and their application to the recent DARPA TIDES-2003 Surprise Language Exercise for Hindi. For comparison, we also implemented an alternative method for surrogate generation: a system that produces topic lists for (Hindi) articles. We present the results of a series of experiments comparing each of these approaches. We demonstrate in both automatic and human evaluations that our linguistically motivated approach outperforms two other surrogate-generation methods: a statistical system and a topic discovery system.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*machine translation*; *text analysis*; *language generation*; H.3.3 [**Information Systems**]: Information Storage and Retrieval—*selection process*

General Terms: Headline Generation, Text Summarization, Cross-Language Information Retrieval

1. INTRODUCTION

There has been a great deal of recent interest in the problem of providing searchers with tools that are tuned to the tasks performed during a cross-language search process. One such task is the recognition of documents relevant to an English query in a non-English collection. This task is difficult because the user must make decisions about relevance based on (often poor) translations of retrieved documents. To facilitate the decision-making process we need translations that can be assessed rapidly and accurately; our approach is to provide an English headline for the non-English document. This was the basis of a series of experiments we ran for the

This work has been supported in part by DARPA TIDES Cooperative Agreement N66001-00-2-8910, BBNT Contract 9500004957, and and NSF CISE Research Infrastructure Award EIA0130422.
Authors' addresses: David M. Zajic, Bonnie J. Dorr, UMIACS, A.V. Williams Building, University of Maryland, College Park, MD 20742; and
Richard Schwartz, BBN Technologies, 9861 Broken Land Parkway, Suite 156, Columbia, Maryland 21046.
E-mails: bonnie, dmzajic@umiacs.umd.edu schwartz@bbn.com

recent DARPA TIDES-2003 Surprise Language Exercise for Hindi.

Specifically, our focus is on generation of headlines for English newspaper texts, with an eye toward the production of document surrogates for cross-language information retrieval.[1] We describe two implemented systems for producing English headlines from translated Hindi articles, one is based on statistics and the other is based on linguistically motivated heuristics. For comparison, we also implemented an alternative method for surrogate generation based on topic lists for (Hindi) articles produced by BBN's Unsupervised Topic Discovery [Schwartz et al. 2001] and OnTopic^{TM}[Schwartz et al. 1999].

The next section describes the background for the headline generation task. In Section 3, we describe our overall approach and then present the framework for each of our headline-generation systems. Following this, we describe our Hindi surprise-language experiments and present the results.

2. BACKGROUND

Headline generation is a form of text summarization in which the summary is required to be informative and very short. Newspaper articles are usually associated with a headline. Headlines are written by copy editors after an article is complete. The copy editors try to construct headlines that satisfy three goals: to summarize the story, to draw in the reader and to fit in the specified space. [Rooney and Witte 2000]. In order to achieve these goals, headline writers adopt a form of compressed English, sometimes referred to as Headlinese [Mårdh 1980]. Some differences between Headlinese and standard English are the omission of determiners and forms of the verb "to be", and use of present tense for events in the past. A headline that summarizes a story is an *informative* abstract, whereas a headline that identifies the topic or topics of a story is an *indicative* abstract. Automatic generation of informative headlines is an important goal because there are documents for which an informative headline in English is not provided, such as non-English articles and transcriptions of broadcast news.

One approach to producing headline-like surrogates is that of sentence compression [Knight and Marcu 2000], whereby a single sentence is shortened using statistical compression. Our approach is similar to this in that we select headline words from story words in the order that they appear in the story. However, in one of our two approaches, we use linguistically motivated heuristics for shortening the sentence; there is no statistical model, which means we do not require any prior training on a large corpus of story/headline pairs.

Other researchers have investigated the topic of automatic generation of abstracts, but the focus has been different, e.g., sentence extraction [Edmundson 1969]; [Johnson et al. 1993]; [Kupiec et al. 1995]; [Mann et al. 1992]; [Teufel and Moens 1997]; [Zechner 1995], processing of structured templates [Paice and Jones 1993], sentence compression [Knight and Marcu 2000]; [Luhn 1958], and generation of abstracts from multiple sources [Radev and McKeown 1998]. We focus instead on the construction of headline-style abstracts from a single story.

[1] Ultimately, we plan to use the same approach for generation of readable headlines from non-English speech broadcasts.

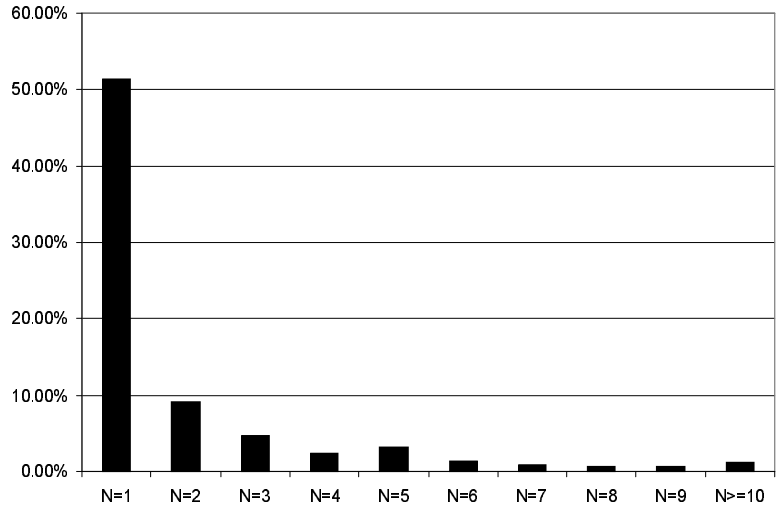

Fig. 1. Percentage of words from human-generated headlines drawn from the Nth sentence of story

3. APPROACH

Our general approach to headline generation is to select words in order from the text of the story. To determine the feasibility of this approach we attempted to apply the technique by hand. We examined 56 English news stories randomly chosen from the TIPSTER corpus. Taking hand-selected story words in the order in which they appeared we were able to construct fluent and accurate headlines for 53 of the stories. The remaining 3 stories were a list of commodity prices, a chronology of events and list of entertainment events. From this we concluded that this approach has promise for stories that are written as paragraphs of prose. We also performed a study in which two subjects were asked to write headlines for 73 AP stories from the TIPSTER corpus for January 1, 1989 by selecting words in order from the story. Of the 146 headlines, 2 did not meet the story-words-in-order criterion because of accidental word reordering. At least one fluent and accurate headline meeting the criterion was created for each of the stories. The average length of the headlines was 10.76 words.

Another way to examine the results is to consider the distribution of the headline words among the sentences of the stories, i.e. how many came from the first sentence of a story, how many from the second sentence, etc. The results of this study on the headlines generated for the 73 AP stories for January 1, 1989 are shown in Figure 1. 51.5% of the headline words were chosen from the first sentence of the story.

Although humans do not always select headline words from the first sentence, we observe that a large percentage of headline words are often found in the first sentence, and that the incidence of headline words chosen from sentences trails off quickly as the sentences are farther into the story. This coincides with the informal observation that news stories are often written with a lead sentence and lead paragraph that summarize the story.

Consider the following excerpt from a news story and corresponding headline.

(1) After months of debate following the Sept. 11 terrorist highjackings, the Transportation Department has decided that airline **pilots** will **not** be **allowed to have guns in** the **cockpits**.

(2) Pilots not allowed to have guns in cockpits.

The bold words in (1) form a fluent and accurate headline, as shown in (2).

This basic approach has been realized in two ways. The first, HMM Hedge, uses a method analogous to Statistical MT to find the most likely headline for a given story. The second, Hedge Trimmer, uses empirically-motivated heuristics to remove grammatical constituents from the lead sentence until it meets a shortness requirement. Both systems have been ported to the cross-language application of English headline generation from Hindi documents.

3.1 Statistical Headline Generation

HMM Hedge (Hidden Markov Model HEaDline GEnerator) is a statistical approach to headline generation. This approach is similar to statistical machine translation in that the observed story is treated as the garbled version of an unseen headline transmitted through a noisy channel. The noisy-channel approach has been used for a wide range of Natural Language Processing (NLP) applications including speech recognition [Bahl et al. 1983], machine translation [Brown et al. 1990], sentence boundary detection [Gotoh and Reynolds 2000], spelling correction [Mays et al. 1990], language identification [Dunning 1994], part-of-speech tagging [Cutting et al. 1992], syntactic parsing [Collins 1997b] [Charniak 1997], semantic clustering [Lin 1998] [Pereira et al. 1993], sentence generation [Langkilde and Knight 1998] [Bangalore and Rambow 2000], and text summarization [Knight and Marcu 2000]. We apply a similar technique to a new domain: automatic generation of headlines from stories.

The intuition behind the algorithm is to treat the observed data (the story) as the result of unobserved data (headlines) that have been distorted by transmission through a noisy channel. The effect of the noisy channel is to add story words between the headline words.

In our cross-language experiments, the English headline words are changed into their Hindi translations in the the story language. The task is to find the English headline most likely to have generated a given Hindi story. That is, each Hindi story word is taken to be generated either from an English headline word or from a general Hindi story language model. Thus stories consist of the story-language translation of headline words with many other story-language words interspersed amongst them.

Formally, if H is an ordered subset of English translations of the first N words of Hindi story S, we want to find the English headline H which maximizes the likelihood that H generated the Hindi story S, or:

$$argmax_H P(H|S)$$

It is difficult to estimate $P(H|S)$, but this probability can be expressed in terms of

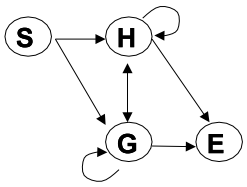

Fig. 2. Simplest HMM to generate stories from headlines

other probabilities that are easier to compute, using Bayes' rule:

$$P(H|S) = \frac{P(S|H)P(H)}{P(S)}$$

Since the goal is to maximize this expression over H, and $P(S)$ is constant with respect to H, $P(S)$ can be omitted. Thus we wish to find:

$$argmax_H P(S|H)P(H)$$

Let H be an English headline consisting of words $h_1, h_2, ..., h_n$. The special symbols *start* and *end* represent the beginning and end of a headline. $P(H)$ is estimated using the bigram probabilities of the words in the headlines:

$$P(H) = P(h_1|start)P(h_2|h_1)...P(h_n|end)$$

The bigram probabilities of the English headline words were calculated from a corpus of 714,184 English headlines from the TIPSTER corpus. The headlines contain 8,692,181 words from a vocabulary of 146,702 distinct words.

To estimate $P(S|H)$ we must consider the process by which an English headline generates a Hindi story. This process can be represented by a Hidden Markov Model (HMM). A HMM is a weighted finite-state automaton in which each state probabilistically emits a string. The simplest HMM for generating headlines is shown in Figure 2.

For simplicity, we consider the monolingual case first. For any story, the HMM consists of a start state S, end state E, an H state for each word in the story, and a corresponding G state for each H state. An H state can emit only the word itself. The corresponding G state remembers which word was emitted by its H state and can emit any word in the story language. A headline corresponds to a path through the HMM from S to E that emits all the words in the story in the correct order. In practice the HMM is constructed with states for only the first N words of the story, where N is a constant (60), or N is the number of words in the first sentence.[2]

In the monolingual example (1) above, the H state will emit the words in bold (pilots, not, allowed, to, have, guns, in, cockpits), and the G state will emit all the other words. The HMM will transition between the H and G state as needed to generate the words of the story. In the current example, the HMM will have states $Start, G_{start}, End$ and 28 H states with 28 corresponding G states. The headline

[2]Limiting consideration to the early part of the story is justified by the observations diagrammed above in Figure 1. Other methods for selecting the window of story words are possible and will be explored in future research.

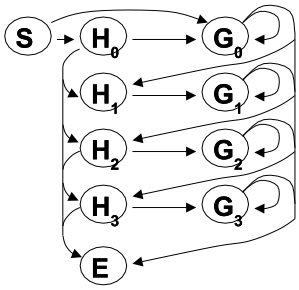

Fig. 3. HMM for a three-word story

given in example (2) corresponds to the following sequence of states: $Start$, G_{start} 18 times, S_{pilots}, G_{pilots}, S_{not}, G_{not}, $S_{allowed}$, S_{to}, S_{have}, S_{guns}, S_{in}, G_{in}, $S_{cockpits}$, End. This path is not the only one that could generate the story in (1). Other possibilities are:

(3) Transportation Department decided airline pilots not to have guns.

(4) Months of the terrorist has to have cockpits.

Although (3) and (4) are possible headlines for (1), the conditional probability of (4) given (1) will be lower than the conditional probability of (3) given (1).

Because a bigram model of headlines is used, the HMM in Figure 2 will not be sufficient. For unconstrained headlines there would have to be an H state for each word in the headline vocabulary. However, because the headline words are chosen from the story words, it will be sufficient to have an H state for each word in the story. Each H state will have a corresponding G state which emits (non-headline) story words until the next headline word and remembers the previously emitted headline word.[3] The HMM for a three-word story is shown in Figure 3.

To port this English headline-generation approach to Hindi, we allowed for the production of multiple translations of Hindi words. This is handled by creating multiple H states (for English headline words) with corresponding G states (for Hindi story words). For instance, the Hindi word *sa.nvidhaana* can be translated to English as *constitution* or *constituent*. Thus, there will be two H states capable of emitting *sa.nvidhaana*, corresponding to headline words *constitution* and *constituent*

The story language is represented by a unigram model calculated from a corpus of 2976 Hindi stories from the BBC corpus. The stories were translated from the original Devanagari into ITRANS.[4] The stories contain 1,184,603 words from a vocabulary of 56,369 distinct words.

The English translations of Hindi words were taken from a scored Hindi-English lexicon produced by ISI during the surprise-language exercise. $P(S|H)$ is the probability of the Hindi story words that are inserted amongst the English headline words. For a given story and headline, if we let $W = w_1, w_2, ..., w_m$ be the words from the story which are not in the headline, and $P(w_i)$ be the unigram probability

[3]A G state can emit any word in the story language model. The story language is represented by a unigram model.

[4]ITRANS is an ascii character transliteration of Hindi which is readable by native Hindi speakers.

in the story language of w_i then

$$P(S|H) = P(w_1)P(w_2)...P(w_m)$$

The Viterbi algorithm is used to select the most likely English headline for a given Hindi story. A two dimensional array of cells is constructed with a row for each state in the HMM and a column for each word in the observed story. Each cell contains a log probability and a backtrace to a cell in the previous column. The cells in the first column are initialized so that the log probability of the start state is 0 and all others are negative infinity. The subsequent columns are filled in with reference to the contents of the previous column.

The H state cells are assigned as follows. For each H state in the previous column, add to the log probability in that cell the log probability that the current story word follows the headline word emitted by that H state in the headline language model. For each G state in the previous column, add to the log probability in that cell the log probability that the current story word follows the headline word emitted by the H state corresponding to that G state. Then select the highest log probability to store in the current cell and include a backtrace to the cell in the previous column from which that log probability was calculated.

The G state cells are assigned as follows. There are only two states in the previous column which can transition to a given G state: the corresponding H state and the G state itself. We select the one with the highest log probability and add to it the log probability that the current story word is generated by the story model, and set the backtrace. Once the final column is filled in, we follow the backtraces from the cell for the end state and end symbol. The resulting headline includes only those words emitted by H states. Finally, to get multiple possible headlines at each length, the algorithm is augmented to include the n-best back-pointers for each partial headline length up to that state. This allows the system to produce many possible headlines which can then be scored by global measures, such as how well the headline can be parsed.

We observed that stories had inherent length biases. Some stories favored longer headlines while others favored shorter headlines. While we do not have an explanation for it at this time, we have accommodated for this phenomenon by adjusting the scores of the best headlines at each length by the slope of the least-squares fit line of the plot of length vs. score. In the case of Hindi to English headline generation there was a strong bias in favor of short headlines when we used this length adjustment, so we implemented a run in which the length bias was forced to favor longer headlines.

A final extension to HMM Hedge is the addition of three decoding penalties to help the system choose the headlines that best mimic actual headlines: a position penalty, a string penalty and a gap penalty.[5] The three penalties were motivated by intuitive observations of the output and their values were set by trial and error. A logical extension to this work would be to attempt to learn the best setting of these penalties, e.g., through Expectation Maximization [Collins 1997a].

Based on our observations of human-constructed headlines, where headline words

[5]The incorporation of these penalties changes the values in the cells from log probabilities to relative desirability scores.

tended to appear near the front of the story, the position penalty is used to favor headlines which include story words near the front of the story. The initial position penalty p is a positive number less than one. The story word in the nth position is assigned a position penalty of $log(p^n)$. When an H state emits a story word, the position penalty is added to the desirability score. Thus words near the front of the story carry less of a position penalty than words farther along. This generalization doesn't hold in the case of human interest and sports stories that start with a hook to get the reader's attention, rather than a topic sentence.

We also observed that in the human-constructed headlines, there were often contiguous strings of story words in the headlines. Example (2) illustrates this with the string "allowed to have guns." The string penalty is used as a bias for "clumpiness", i.e., the tendency to generate headlines composed of strings of contiguous story words. The log of the string penalty is added to the desirability score with each transition from an H state to its G state. High string penalties cause a bias towards headlines consisting of fewer but larger clumps of contiguous story words.

The gap penalty is used to bias against headline "gappiness", i.e. large gaps of non-headline words in the story between clumps of headline words. Although humans were capable of constructing fluent headlines by selecting widely spaced words, we observed that the algorithm was more likely to combine unrelated material by doing this. At each transition from a G state to an H state, corresponding to the end of a sequence of non-headline words in the story, a gap penalty is applied which increases with the size of the gap since the last headline word was emitted. This can also be seen as a penalty for spending too much time in one G state. High gap penalties cause a bias for headlines with few large gaps.

3.2 Parse-and-Trim Headline Generation

Our second approach to constructing headlines also involves selecting words in order from a story. This approach, called Hedge Trimmer, involves removal of grammatical constituents from a parse of the lead sentence until a length threshold has been met. The parses are created by the BBN SIFT parser. As described in [Miller et al. 1998] the BBN SIFT parser builds augmented parse trees according to a process similar to that described in [Collins 1997c]. The BBN SIFT parser has been used successfully for the task of information extraction [Miller et al. 2000].

The approach taken by Hedge Trimmer is most similar to that of [Knight and Marcu 2000], where a single sentence is shortened using statistical compression. However, Hedge Trimmer uses linguistically motivated heuristics for shortening the sentence. There is no statistical model, so prior training on a large corpus of stories and headlines is not required.

The input to Hedge Trimmer is a story. The first sentence of the story is passed through the BBN SIFT parser. The parse-tree result serves as input to a linguistically motivated module that selects story words to form headlines based on key insights gained from observations of human-constructed headlines.

We produced English headlines for the 25 surprise language evaluation stories by translating the stories into English using the statistical Hindi-English Machine Translation system developed by ISI during the surprise language exercise. Once the story was translated into English, we ran the first sentence through the Hedge Trimmer system. While we might have implemented an algorithm to select the best

Level	Phenomenon	Count	Percentage
Headline	preposed adjuncts	0/212	0%
Headline	conjoined S	1/218	0.5%
Headline	conjoined VP	7/218	3%
Noun Phrase	relative clause	3/957	0.3%
Noun Phrase	determiner	31/957	3%
Clause	time expression	5/315	1.5%
Clause	trailing PP	165/315	52%
Clause	trailing SBAR	24/315	8%

Table I. Percentages found in human-generated headlines

Level	Phenomenon	Count	Percentage
Headline	preposed adjuncts	2/73	2.7%
Headline	conjoined S	3/73	4%
Headline	conjoined VP	20/73	27%
Noun Phrase	relative clause	29/817	3.5%
Noun Phrase	determiner	205/817	25%
Clause	time expression	77/316	24%
Clause	trailing PP	184/316	58%
Clause	trailing SBAR	49/316	16%

Table II. Percentages found in story first sentences

sentence to trim, we did not have time and found that, for news stories, the first sentence is often the best. The remainder of this section describes the linguistic justifications for and operation of the Hedge Trimmer system.

Our observations about human headlines seem to coincide with several of the choices we made for removing constituents from the parse tree. To verify this, we ran 218 human-constructed headlines through the BBN SIFT parser and compared the resulting structures to the SIFT-parsed first sentence of each story.

For the 218 human-constructed headlines, the parser produced 957 noun phrases (NP) and 315 clauses (S). At each level (headline, noun phrase and clause), different types of linguistic phenomena were counted. At headline level, the number of headlines containing preposed adjuncts, conjoined clauses and conjoined verb phrases were counted. At the NP level, the number of NPs containing determiners and relative clauses were counted. At the S level, the number of clauses containing time expressions, trailing SBARs and trailing PPs were counted. The counts and percentages of the phenomena in the human-generated headlines are shown in Table I.

For comparison, the same phenomena were counted for parses of the first sentences of 73 AP stories from the TIPSTER corpus for January 1, 1989. The parser results included 817 noun phrases and 316 clauses. The counts and percentages of the phenomena in the first sentences of stories are shown in Table II.

The comparison of the prevalence of these phenomena in human-generated headlines to story first sentences suggests that they are reasonable choices for trimming, with the exception of trailing PPs. Thus special care is taken to remove PPs late in

the process and to reduce the likelihood of removing a PP that contains important content.

Hedge Trimmer uses the following algorithm for parse-tree trimming:

Step 1. Choose the lowest leftmost S with NP, VP
Step 2. Remove low content units
  (a) Some determiners
  (b) Time expressions
Step 3. Iterative shortening
  (a) XP-over-XP reduction
  (b) Remove preposed adjuncts of root S
  (c) Remove trailing PPs
  (d) Remove trailing SBARs

The first step, which chooses an appropriate S node, corresponds to the *Projection Principle* in Linguistic theory [Chomsky 1981]: predicates project a subject (directly dominated by S) in the surface structure. The human-generated headlines we studied always conformed to this rule. Thus it has been adopted as a constraint in the Hedge Trimmer algorithm that the lowest leftmost S node which has as children both a NP node and a VP node in that order is taken to be the root node of the headline. An example of the application of this step is shown in (5). The boldfaced material in the parse is retained and the italicized material is eliminated.

(5) Input: Rebels agreed to talks with government officials, international observers said Tuesday.
Parse: *[S* [**S** [**NP Rebels**][**VP agreed to talks with government officials**]], *international observers said Tuesday.]*
Output: Rebels agreed to talks with government officials.

When the parser produces a correct tree, this step provides a grammatical headline. However, the parser often produces incorrect output. When the parser was run on the 624-sentence DUC-2003 evaluation set, human evaluation of the output revealed that there were two such scenarios.

(6) Parse: [**S**[**SBAR What started as a local controversy**][**VP has evolved into an international scandal.**]]

(7) Parse: [**NP**[**NP Bangladesh**][**CC and**][**NP**[**NP India**][**VP signed a water sharing accord.**]]]

In (6), an S exists, but it does not conform to the requirements of the Projection Principle because it does not have as children a NP followed by a VP. This occurred in 2.6% of the sentences in the DUC-2003 evaluation data. The problem is resolved by selecting the lowest leftmost S, whether or not it is the parent of an NP followed by a VP.

In (7), no S is present in the parse. This occurred in 3.4% of the sentences in the DUC-2003 evaluation data. This problem is resolved by selecting the root of the entire parse tree as the root of the headline.

Step 2 of the algorithm removes low-content units from the parse tree. The simplest low-content units are the determiners *a* and *the*. Other determiners are

not considered for deletion because the analysis of the human-constructed headlines revealed that most of the other determiners provide important information, e.g., negation (not), quantifiers (each, many, several), and deictics (this, that).

In addition, time expressions—although certainly not content-free—are not vital for summarizing the theme of an article. Since the goal is to provide an informative headline, the identification and elimination of time expressions allows other more important details to remain in the length-constrained headline. Time expressions are identified with BBN's IdentiFinder^{TM}[Bikel et al. 1999]. The elimination of time expressions is a two-step process: (a) Use IdentiFinder^{TM}to mark time expressions; and (b) Remove [PP ... [NP [X] ...] ...] and [NP [X]] where X is tagged as part of a time expression.

The following examples illustrate the application of time expression removal:

(8) Input: The State Department on Friday lifted the ban it had imposed on foreign fliers.
Parse: [**S** [**NP***[Det The]* **State Department** *[PP [IN on] [NP [NNP Friday]]]* [**VP lifted** *[Det the]* **ban it had imposed on foreign fliers.**]]
Output: State Department lifted ban it had imposed on foreign fliers.

(9) Input: An international relief agency announced Wednesday that it is withdrawing from North Korea.
Parse: [**S** [**NP** *[Det An]***international relief agency**][**VP announced** *[NP [NNP Wednesday]]* **that it is withdrawing from North Korea.**]]
Output: International relief agency announced that it is withdrawing from North Korea.

53.2% of the first sentences in the DUC-2003 evaluation data contained at least one time expression that could be removed. Human inspection of 50 deleted time expressions showed that 38 were desirable deletions, 10 were locally undesirable because they introduced an ungrammatical fragment, and 2 were undesirable because they removed a potentially relevant constituent. However, even an undesirable deletion often pans out for two reasons: (1) the ungrammatical fragment is frequently deleted later by some other rule; and (2) every time a constituent is removed it makes room under the threshold for some other, possibly more relevant constituent. Consider the following examples, created by Hedge Trimmer with a threshold of 20 words.

(10) New York Times said in editorial for Friday, Sept 10 naming of former Sen. John Danforth is promising development

(11) New York Times said in editorial naming of former Sen. John Danforth to conduct independent inquiry Texas is promising development.

Headline (10) was produced by a system which did not remove time expressions. Headline (11) shows that if the 4-word time expression *for Friday, Sept 10* were removed, it would make room below the 20-word threshold for another important piece of information, i.e., *to conduct independent inquiry Texas*.

The final step of the algorithm, iterative shortening, removes linguistically peripheral material through successive deletions of constituents until the sentence is

shorter than a given threshold. The headline length threshold is a configurable parameter. There are four types of iterative shortening. For each type of shortening, the positions in the parse tree where it is possible to apply the shortening rule are found, and then the shortening rule is applied to those positions from the deepest, rightmost back until the headline is under the length threshold. When a shortening rule has been applied at all possible places in the parse tree and the headline is still above the length threshold, the algorithm moves to the next type of shortening rule. The four shortening rules are: (a) XP-over-XP Reduction; (b) Remove preamble of root S; (c) Remove trailing PPs; and (d) Remove trailing SBARs.

XP-over-XP reduction is implemented as follows: In constructions of the form [XP [XP ...] ...] remove the other children of the higher XP, where XP is NP, VP or S. This is a linguistic generalization that allows the application of a single rule to capture three different phenomena: relative clauses, verb-phrase conjunction and sentential conjunction. The rule is applied iteratively, from the deepest rightmost applicable node backwards, until the length threshold is reached. The impact of XP-over-XP reduction can be seen in these examples of NP-over-NP (relative clauses), VP-over-VP (verb-phrase conjunction) and S-over-S (sentential conjunction), respectively.

(12) Input: A fire killed a firefighter who was fatally injured as he searched the house.
Parse: [**S***[Det A]***fire killed***[Det a]*[**NP** [**NP firefighter***[SBAR who was fatally injured as he searched the house.]*]]]
Output: Fire killed firefighter.

(13) Input: Illegal fireworks injured hundreds of people and started six fires.
Parse: [**S Illegal fireworks** [**VP** [**VP injured hundreds of people**]*[CC and] [VP started six fires.]*]]
Output: Illegal fireworks injured hundreds of people.

(14) Input: A company offering blood cholesterol tests in grocery stores says medical technology has outpaced state laws, but the state says the company doesn't have the proper licenses.
Parse: [**S***[Det A]***company offering blood cholesterol tests in grocery stores says** [**S** [**S medical technology has outpaced state laws,**]*[CC but] [S the state says the company doesn't have the proper licenses.]*]]
Output: Company offering blood cholesterol tests in grocery stores says medical technology has outpaced state laws.

The motivation for removal of preposed adjuncts is that all of the human-generated headlines ignored what we refer to as the *preamble* of the story. Assuming the Projection Principle has been satisfied, the preamble is viewed as the phrasal material occurring before the subject of the sentence. Thus, adjuncts are identified linguistically as any XP unit preceding the first NP (the subject) under the S chosen by step 1. Note that this step is not iterative, but it is included here because it is only applied if the first step of iterative shortening has not reduced the headline

below the threshold length. The impact of preposed adjunct removal can be seen in example (15).

(15) Input: According to a now finalized blueprint described by U.S. officials and other sources, the Bush administration plans to take complete, unilateral control of a post-Saddam Hussein Iraq.
Parse: [**S***[PP According to a now finalized blueprint described by U.S. officials and other sources], [Det the]***Bush administration plans to take complete, unilateral control of***[Det a]***post-Saddam Hussein Iraq.**]
Output: Bush administration plans to take complete unilateral control of post-Saddam Hussein Iraq.

The third and fourth types of iterative shortening are the removal of trailing PPs and SBARs, respectively. These are the riskiest of the iterative shortening rules, as indicated in the analysis of the human-generated headlines. Thus, these rules are applied last, only when there are no other categories of rules to apply. Moreover, these rules are applied with a backoff option to avoid over-trimming the parse tree. First the PP shortening rule is applied. If the threshold has been reached, no more shortening is done. However, if the threshold has not been reached, the system reverts to the parse tree as it was before any PPs were removed, and applies the SBAR shortening rule. If the threshold still has not been reached, the PP rule is applied to the output of the SBAR rule. The intuition is that when removing constituents from a parse tree, it's best to remove smaller portions during each iteration to avoid producing trees with very few words. PPs tend to represent small parts of the tree while SBARs represent large parts of the tree. Thus we try to reach the threshold by removing small constituents, but if we can't reach the threshold that way, we restore the small constituents, remove a large constituent and resume the deletion of small constituents.

In an effort to reduce the risk of removing PPs containing important information, BBN's IdentiFinder^{TM}is used to distinguish PPs containing a named entity. PPs containing named entities are not removed during the first round of PP removal. However, PPs containing named entities that are descendents of SBARs are removed before the parent SBAR is removed. The reason is that we should try to reach the threshold by removing a small constituent before removing a larger constituent that subsumes it. The impact of these two types of shortening can be seen in examples (16) and (17).

(16) Input: More oil-covered sea birds were found over the weekend.
Parse: [**S More oil-covered sea birds were found**]*[PP over the weekend]*]
Output: More oil-covered sea birds were found.

(17) Input: Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition while assuring closer cooperation after Hong Kong returns.
Parse: [**S Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition***[SBAR while assuring closer cooperation after Hong Kong returns.]*]
Output: Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition.

Other sequences of shortening rules are possible. The one above was observed to produce the best results on a 73-sentence development set of stories from the TIPSTER corpus.

At present, Hedge Trimmer is applied to the problem of cross-language headline generation by translating the first sentence of a story into English and running the Hedge Trimmer process on the resulting translation. The obvious drawback is that it requires a translation process.

3.3 Unsupervised Topic Discovery

BBN's OnTopic^{TM}system uses a HMM to assign topics from a topic-annotated corpus to a new document. The system assumes that documents are generated by a set of topics that emit topic words and a general language model that emits all other words. The OnTopic^{TM}system finds the most likely set of topics for a document. For example, the following list might be a typical set of topics: "Bombings", "Terrorism", "Oklahoma" and "Deaths and Injuries".

However, we often do not have a corpus annotated with topics, especially for a new language. Thus we use an algorithm for Unsupervised Topic Discovery (UTD) that takes as input a large unannotated corpus in any language and automatically creates a set of topic models with meaningful names. The algorithm has several stages. First, it analyzes the corpus to find strings of words that occur frequently. (It does this using a Minimum Description Length criterion.) These are frequently phrases that are meaningful names of topics. Second, it finds the high-content words in each document (using a modified tf.idf measure). These are possible topic names for each document. It keeps only those names that occur in at least four different documents. These are taken to be an initial set of topic names. The third stage is to train topic models corresponding to these topic names. The modified EM procedure of OnTopic^{TM}is used to determine which words in the documents often signify these topic names. This produces topic models. Fourth, these topic models are used to find the most likely topics for each document. This often adds new topics to documents, even though the topic name did not appear in the document.

We found, in various experiments, that the topics derived by this procedure were usually meaningful and that the topic assignment was about as good as when the topics were derived from a corpus that was annotated by people. We have also used this procedure on different languages and shown the same behavior.

Note that the UTD algorithm does not need to understand the words. They are just character strings. UTD can run equally well on a new language as long as it can be divided into strings that approximate words.

To find topics for the Hindi stories, we first ran the UTD algorithm on a corpus of 1M words of Hindi stories from the BBC Hindi corpus. (A native speaker provided a list of stop words.) This produced a set of Hindi topics. Then, the native speaker examined the topics produced and rejected the proposed topics that could not serve as topics (e.g. particles, conjunctions and numbers), and provided English translations of the valid UTD topics. OnTopic^{TM}was used to assign topics to the test documents.

System	Average Headline Length (in words)
STAT1	8.36
STAT2	11.12
TRIMMER	12.36
UTD	12.68

Table III. Average Headline Lengths

4. EVALUATION

We evaluated the results of the two headline-generation systems, comparing them to the UTD approach. To determine our degree of effectiveness, we used two automatic methods, BLEU [Papineni et al. 2002] and ROUGE [Lin and Hovy 2003], as well as a manual evaluation based on human judgments of clarity. In both types of evaluations we used four human-generated reference headlines for each story provided by NIST in the Hindi Surprise Language Exercise. We evaluated four systems: Hedge Trimmer, two variants of HMM Hedge (STAT1 and STAT2) and UTD. The difference between STAT1 and STAT2 is that STAT2 was biased to favor longer headlines. The average length of the headlines produced by the systems is shown in Table III. UTD produced an average of 9.10 topics per document, however many of the topics are multi-word topics, and the average number of words was 12.68. Hedge Trimmer was run with a length threshold of 15 to produce headlines with average length 12.36.

BLEU is a system for automatic evaluation of machine translation that uses a modified n-gram precision measure to compare machine translations to reference human translations. This automatic metric counts the number of n-grams in the candidate that occur in any of the reference summaries and divides by the number of n-grams in the candidate. In our evaluation of headline generation systems, we treat summarization as a type of translation from a verbose language to a concise one, and compare automatically generated headlines to human generated headlines.

Figure 4 shows the BLEU scores of our four headline-generation systems. In these experiments, BLEU was configured to use unigrams through 4-grams. The BLEU scores show with that Hedge Trimmer scored better than the other systems; that there was no substantial difference between the two statistical systems; and that UTD topic lists scored worse than the other systems.

A second automatic evaluation was performed using the ROUGE metric for evaluation of text summarization. ROUGE is a recall-based measure, analogous to BLEU. This automatic metric counts the number of n-grams in the reference summaries that occur in the candidate and divides by the number of n-grams in the reference summaries. Figure 5 shows the ROUGE scores of our four systems. As with BLEU, ROUGE was configured to use unigrams through 4-grams. As in the BLEU evaluation, the ROUGE scores show that Hedge Trimmer performs better than the other systems across N, and that there is no substantial difference between the two statistical systems. However, at N=1, UTD scores better than the statistical systems. At N>1, there is no difference among the STAT and UTD systems.

Our final evaluation involved human clarity judgments for the output of STAT2,

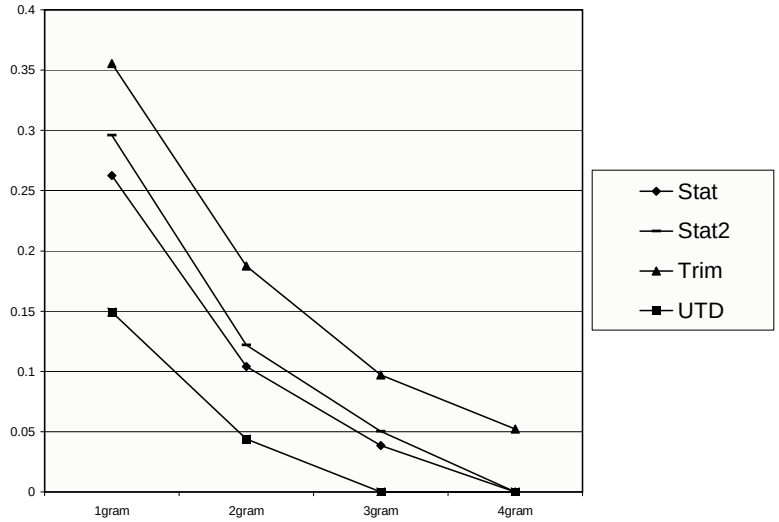

Fig. 4. BLEU Scores for UMD/BBN systems on Hindi Headlines

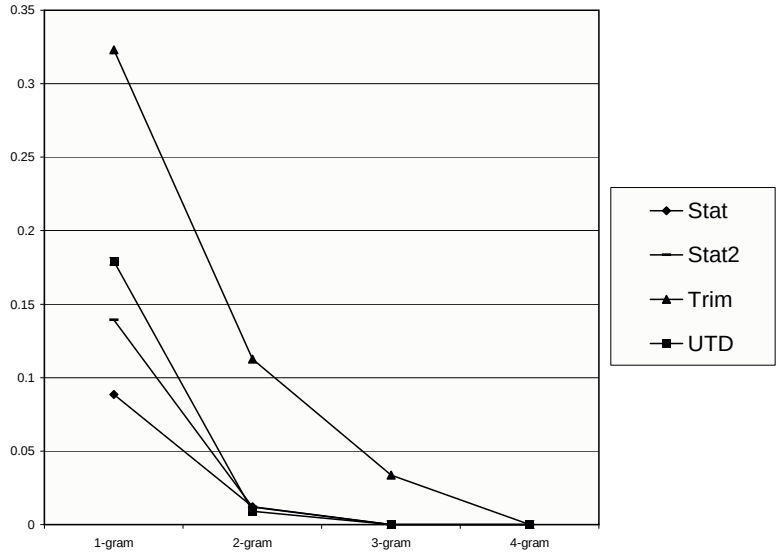

Fig. 5. ROUGE Scores for UMD/BBN systems on Hindi Headlines

Hedge Trimmer and UTD topic lists. We showed three subjects the output of the three systems and asked them to rank the summaries from 1 (worst) to 5 (best) for whether they could tell what happened in the story based on the headline. We then showed the subjects the four reference summaries and asked the subjects to rank the headlines on the same scale whether the summaries were correct. The results of this evaluation are shown in Figure 6. We found that the human judgments coincided with those of both automatic metrics: Hedge Trimmer performed better

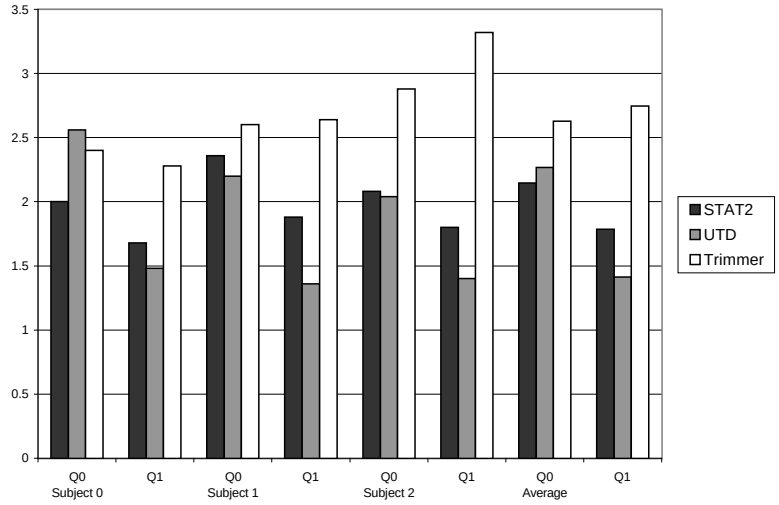

Fig. 6. Human Evaluation of UMD systems for Hindi Headlines.
Q0: Can you tell what happens in the story based on the candidate summary?
Q1: Based on the reference summaries, was the candidate summary correct?

than the other systems and UTD was substantially lower in all three evaluations.

5. CONCLUSION

We have presented new approaches to *headline generation* of English newspaper texts—and we have extended the approach to operate on Hindi for the TIDES-2003 Surprise Language Exercise. We have demonstrated in both automatic and human evaluations that our linguistically motivated approach (Hedge Trimmer) outperforms two other methods for surrogate generation—a statistical system (HMM Hedge) and a system that produces topic lists (UTD). Moreover, our automatic evaluations correlate with human judgments.

The results of this work suggest several future research directions. One area is that of developing a hybrid version of headline generation that combines the language modeling aspect of the statistical approach with the linguistically-informed heuristics of the symbolic approach. We will explore the question of which techniques perform best with respect to document type and intended use.

Another area of future research is the investigation of the usefulness of headlines for an extrinsic task—and also the targeting of headlines for a particular application. For example, the current versions of our headline-generation systems are not necessarily appropriate for the task of judging the relevance of a document to a particular query. Thus, future research will focus on the use of a wide range of headline-generation techniques for the extrinsic task of interactive selection of relevant documents. There are two goals overall:

(1) To measure usefulness of different types of headlines/summaries for the task of relevance judgment.
(2) To calibrate the automatic evaluation metrics (e.g., ROUGE and BLEU) under several different environments in order that we can use this metric with

confidence, e.g., in future NIST-organized DUC evaluations.

Regarding the first of these goals, we will take several different kinds of measurements. First, for information retrieval, we will compare headlines, short phrases, and topics/keywords to see which is best for judging relevance. While the headlines are clearly more fluent, the topics are more efficient at providing many concepts. Second, we will use the experiments to identify strategies that are best for cross-language summarization, given the higher level of noise in translated stories: headlines, an integrated probabilistic approach, or cross-language topics that require no cross-language resources. Third, we will develop, and demonstrate the benefit of, query-targeted summarization over generic summarization; the aim is to show that query-focused summarization makes it easier for the human to accurately judge document relevance. We will compare summarization with a high baseline, i.e., KWIC (key-word-in-context) as used by Google, but limited to headline-length output.

Regarding the second of these goals, we will use human-generated references to tune automatic metrics in order to maximize the correlation with the measured usefulness, i.e., with the accuracy and speed of the humans judging relevance from the summaries.

Another area of to explore is the development of an approach to selecting the lead sentence. Currently, we select the first sentence of the story as the lead sentence. However, other methods of selecting the window of story words are possible (e.g., using tf.idf to detect terms with high information content) will be explored in future research.

Finally, we plan to apply both systems to speech transcriptions of broadcast news and conversations. Determining which window of a speech transcript will be a significant issue, because we do not expect broadcast news to follow the same discourse patterns as written news articles. Also, speech transcripts will be much noisier than written text, because of errors in the transcription and speech disfluencies in the source. Determination of sentence boundaries is also an important issue. As with cross-language headline generation, it is an open question which techniques will perform best for transcribed speech.

REFERENCES

BAHL, L., JELINEK, F., AND MERCER, R. 1983. A maximum likelihood approach to speech recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. PAMI-5(2). 179–190.

BANGALORE, S. AND RAMBOW, O. 2000. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of COLING 2000*.

BIKEL, D., SCHWARTZ, R., AND WEISCHEDEL, R. 1999. An algorithm that learns what's in a name. *Machine Learning 34,* 1/3.

BROWN, P., COCKE, J., PIETRA, S., PIETRA, V., JELINEK, F., LAFFERTY, J., MERCER, R., AND ROOSSIN, P. 1990. A statistical approach to machine translation. *Computational Linguistics 16(2)*, 79–85.

CHARNIAK, M. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI-97*.

CHOMSKY, N. A. 1981. *Lectures on Government and Binding*. Foris Publications, Dordrecht, Holland.

COLLINS, M. 1997a. The EM Algorithm (In fulfillment of the Written Preliminary Exam II requirement).

COLLINS, M. 1997b. Three Generative, Lexicalised Models for Statistical Parsing.

COLLINS, M. 1997c. Three generative lexicalised models for statistical parsing. In *Proceedings of the 35th ACL.*

CUTTING, D., PEDERSEN, J., AND SIBUN, P. 1992. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing.* Trento, Italy.

DUNNING, T. 1994. Statistical identification of language. Tech. Rep. Techincal Report MCCS 94-273, New Mexico State University.

EDMUNDSON, H. 1969. New methods in automatic extracting. *Journal of the ACM 16(2).*

GOTOH, Y. AND REYNOLDS, S. 2000. Sentence boundary detection in broadcast speech transcripts. In *Proceedings of the International Speech Communication Association Workshop: Automatic Speech Recognition: Challenges for the New Millenium.* Paris.

JOHNSON, F., PAICE, C., BLACK, W., AND NEAL, A. 1993. The applicaiton of linguistic processing to automatic abstract generation. *Journal of Document and Text Management 1(3)*, 215–42.

KNIGHT, K. AND MARCU, D. 2000. Statistics-based summarization – step one: Sentence compression. In *The 17th National Conference of the American Association for Artificial Intelligence AAAI2000.* Austin, Texas.

KUPIEC, J., PEDERSEN, J., AND CHEN, F. 1995. A trainable document summarizer. In *Proceedings of the 18th ACM-SIGIR Conference.*

LANGKILDE, I. AND KNIGHT, K. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of COLING-ACL.*

LIN, C.-Y. AND HOVY, E. 2003. Automatic Evaluation of Summaries Using N-gram Co-Occurrences Statistics. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics.* Edmonton, Alberta.

LIN, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of Coling/ACL.*

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

MANN, W., MATTHIESEN, C., AND THOMPSON, S. 1992. Rhetorical Strucutre Theory and Text Analysis. *Discoure Description.*

MÅRDH, I. 1980. *Headlinese: On the Grammar of English Front Page Headlines.* Malmo.

MAYS, E., DAMERAU, F., AND MERCER, R. 1990. Context-based spelling correction. In *Proceedings of IBM Natural Language ITL.* France, 517–522.

MILLER, S., CRYSTAL, M., FOX, H., RAMSHAW, L., SCHWARTZ, R., STONE, R., AND WEISCHEDEL, R. 1998. Algorithms that Learn to Extract Information; BBN: Description of the SIFT System as Used for MUC-7. In *Proceedings of the MUC-7.*

MILLER, S., RAMSHAW, L., FOX, H., AND WEISCHEDEL, R. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st Meeting of the North Amererican Chapter of the ACL.* Seattle, WA, 226–233.

PAICE, C. AND JONES, A. 1993. The Identification of important concepts in highly structured technical papers. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in IR.*

PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of Association of Computational Linguistics.* Philadelphia, PA.

PEREIRA, F., TISHBY, N., AND LEE, L. 1993. Distributional clustering of english words. In *Proceedings of 31st ACL.*

RADEV, D. R. AND MCKEOWN, K. R. 1998. Generating Natural Language Summaries from Multiple On-Line Sources. *Computational Linguistics 24(3)*, 469–500.

ROONEY, E. AND WITTE, O. 2000. *Copy Editing for Professionals.* Stipes Publishing Co.

SCHWARTZ, R., IMAI, T., JUBALA, F., NGUYEN, L., AND MAKHOUL, J. 1999. A maximum likelihood model for topic classification of broadcast news. In *Eurospeech-97.* Rhodes, Greece.

SCHWARTZ, R., SISTA, S., AND LEEK, T. R. 2001. Unsupervised topic discovery. In *Proceedings of Workshop on Language Modeling and Information Retrieval*. Pittsburgh, PA, 72–77.

TEUFEL, S. AND MOENS, M. 1997. Sentence extraction as a classification task. In *Proceedings of the Workshop on Intelligent and scalable Text summarization, ACL/EACL*. Madrid, Spain.

ZECHNER, K. 1995. Automatic Text Abstracting by Selecting Relevant Passages. M.S. thesis, Center for Cognitive Science, University of Edinburgh.