HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space

Carlos A. Colmenares*
Google Inc.
Brandschenkestrasse 110
8002 Zurich, Switzerland
crcarlos@google.com

Marina Litvak
Shamoon College
of Engineering
Beer Sheva, Israel
marinal@sce.ac.il

Amin Mantrach **Fabrizio Silvestri**
Yahoo Labs.
Avinguda Diagonal 177
08018 Barcelona, Spain
{amantrach,silvestr}@yahoo-inc.com

Abstract

Automatic headline generation is a sub-task of document summarization with many reported applications. In this study we present a sequence-prediction technique for learning how editors title their news stories. The introduced technique models the problem as a discrete optimization task in a feature-rich space. In this space the global optimum can be found in polynomial time by means of dynamic programming. We train and test our model on an extensive corpus of financial news, and compare it against a number of baselines by using standard metrics from the document summarization domain, as well as some new ones proposed in this work. We also assess the readability and informativeness of the generated titles through human evaluation. The obtained results are very appealing and substantiate the soundness of the approach.

1 Introduction

Document summarization, also known as text summarization, is the process of automatically abridging text documents. Although traditionally the final objective of text summarization is to produce a paragraph or abstract that summarizes a rather large collection of texts (Mani and Maybury, 1999; Das and Martins, 2007; Nenkova and McKeown, 2012), the task of producing a very short summary comprised of 10–15 words has also been broadly studied. There have been many reported practical applications for this endeavor, most notably, efficient web browsing on hand-held devices (Buyukkokten et al., 2001), generation of TV captions (Linke-Ellis, 1999), digitization of newspaper articles that have uninformative headlines (De Kok, 2008), and headline generation in one language based on news stories written in another (Banko et al., 2000; Zajic et al., 2002).

In general terms, a headline of a news article can be defined as a short statement that gives a reader a general idea about the main contents of the story it entitles (Borko and Bernier, 1987; Gattani, 2007). The objective of our study is to develop a novel technique for generating informative headlines for news articles, albeit to conduct experiments we focused on finance articles written in English. In this work we make a number of contributions concerning statistical models for headline generation, training of the models, and their evaluation, specifically:

- We propose a model that learns how an editor generates headlines for news articles, where a headline is regarded as a compression of its article's text. Our model significantly differs from others in the way it represents possible headlines in a feature-rich space. The model tries to learn how humans discern between good and bad compressions. Furthermore, our model can be trained with any monolingual corpus consisting of titled articles, because it does not request special conditions on the headlines' structure or provenance.
- We suggest a slight change of the Margin Infused Relaxed Algorithm (Crammer and Singer, 2003) to fit our model, which yields better empirical results.

* Work done during an internship at Yahoo Labs.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 133–142,
Denver, Colorado, May 31 – June 5, 2015.

- We present a simple and elegant algorithm that runs in polynomial time and finds the global optimum of our objective function. This represents an important advantage of our proposal because many former techniques resort to heuristic-driven search algorithms that are not guaranteed to find the global optimum.

- With the intention of overcoming several problems suffered by traditional metrics for automatically evaluating the quality of proposed headlines, we propose two new evaluation metrics that correlate with ratings given by human annotators.

2 Related work

There has been a significant amount of research about headline generation. As noted by Gattani (2007), it is possible to identify three main trends of techniques broadly employed through different studies:

Rule-based approaches. These methods make use of handcrafted linguistically-based rules for detecting or compressing important parts in a document. They are simple and lightweight, but fail at exploring complex relationships in the text. The most representative model for this group is the Hedge Trimmer (Dorr et al., 2003).

Statistics-based approaches. These methods make use of statistical models for learning correlations between words in headlines and in the articles. The models are fit under supervised learning environments and therefore need large amounts of labelled data. One of the most influential works in this category is the Naïve Bayes approach presented by Banko et al. (2000), and augmented in works such as Jin and Hauptmann (2001; Zajic et al. (2002). The use of statistical models for learning pruning-rules for parse trees has also been studied, the most notable work on this area is presented in Knight and Marcu (2001) and extended by Unno et al. (2006).

Summarization-based approaches. Headlines can be regarded as very short summaries, therefore traditional summarization methods could be adapted for generating one-line compressions; the common trend consists in performing multiple or combined steps of sentence selection and compression (Hajime et al., 2013; Martins and Smith, 2009). The main problem with these approaches is that they make use of techniques that were not initially devised for generating compressions of less than 10% of the original content, which directly affects the quality of the resulting summary (Banko et al., 2000). It is noteworthy to highlight that most of the modern summarization-based techniques opt for generating headlines just by recycling and reordering words present in the article, which also raises the risk of losing or changing the contextual meaning of the reused words (Berger and Mittal, 2000).

An area that deals with a target similar to headline generation is multi-sentence compression, where its objective is to produce a single short phrase that abridges a set of sentences that conform a document. The main difference between both practices is that headline generation is more strict about the length of the generated output, which should consist of about eight tokens (Banko et al., 2000), whereas the latter accepts longer results. One of the most recent and competitive approaches for multi-sentence compression is described by Filippova (2010).

3 Background on sequence prediction

Sequence models have been broadly used for many Natural Language Processing tasks, such as identification of sentence boundaries (Reynar and Ratnaparkhi, 1997), named entity recognition (McCallum and Li, 2003), part of speech tagging (Kupiec, 1992), dependency tree parsing (McDonald et al., 2005), document summarization (Shen et al., 2007), and single-sentence compression (McDonald, 2006; Nomoto, 2007). These models are formalizations of relationships between observed sequences of variables and predicted categories for each one. Mathematically, let $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ be a finite set of possible atomic observations, and let $\mathcal{Y} = \{y_1, y_2, ..., y_M\}$ be a finite set of possible categories that each atomic observation could belong to.

Statistical sequence models try to approximate a probability distribution P with parameters ϕ capable of predicting for any sequence of n observations $\boldsymbol{x} \in \mathcal{X}^n$, and any sequence of assigned categories per observation $\boldsymbol{y} \in \mathcal{Y}^n$, the probability $P(\boldsymbol{y}|\boldsymbol{x}; \phi)$. The final objective of these models is to predict the

most likely sequence of categories $\hat{\boldsymbol{y}} \in \mathcal{Y}^n$ for any arbitrary observation sequence, which can be expressed as:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathcal{Y}^n}{\arg\max}\ P(\boldsymbol{y}|\boldsymbol{x}; \phi)$$

There have been many proposals for modelling the probability distribution P. Some of the most popular proposals are Hidden Markov Models (Rabiner and Juang, 1986), local log-linear classifiers, Maximum Entropy Markov Models (McCallum et al., 2000), and Conditional Random Fields (Lafferty et al., 2001). The following two sections will briefly introduce the latter, together with a widely used improvement of the model.

3.1 Conditional Random Fields (CRF)

As presented by Lafferty et al. (2001), CRF are sequence prediction models where no Markov assumption is made on the sequence of assigned categories $\boldsymbol{y}$, but a factorizable global feature function is used so as to transform the problem into a log-linear model in feature space. Formally, CRF model the probability of a sequence in the following way:

$$P(\boldsymbol{y}|\boldsymbol{x}; \phi) = \frac{exp\{\boldsymbol{w} \cdot \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y})\}}{Z(\boldsymbol{x})}$$

Where $\phi = \{\boldsymbol{w}\}$ and $\boldsymbol{w} \in \mathbb{R}^m$ is a weight vector, $\boldsymbol{F} : \mathcal{X}^n \times \mathcal{Y}^n \rightarrow \mathbb{R}^m$ is a global feature function of m dimensions, and $Z(\boldsymbol{x})$ is a normalization function. Moreover, the global feature function is defined in the following factored way:

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{x}, i, y_{i\text{-}1}, y_i)$$

where $\boldsymbol{f} : \mathcal{X}^* \times \mathbb{N}^+ \times \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^m$ is a local feature function. Due to this definition, it can be shown that the decoding of CRF is equivalent to:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathcal{Y}^n}{\arg\max}\ \boldsymbol{w} \cdot \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y})$$

Which is a linear classification in a feature space. The fact that the local feature function $\boldsymbol{f}$ only depends on the last two assigned categories allows the global optimum of the model to be found by means of a tractable algorithm, whereas otherwise it would be necessary to explore all the $|\mathcal{Y}|^n$ possible solutions.

3.2 CRF with state sequences

Since CRF do not assume independence between assigned categories, it is possible to extend the local feature function for enabling it to keep more information about previous assigned categories and not just the last category. These models are derived from the work on weighted automata and transducers presented in studies such as Mohri et al. (2002). Let $\mathcal{S}$ be a state space, s_0 be a fixed initial empty state, and let function $g : \mathcal{S} \times \mathcal{X}^* \times \mathbb{N}^+ \times \mathcal{Y} \rightarrow \mathcal{S}$ model state transitions. Then the global feature function can be redefined as:

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{x}, i, s_{i\text{-}1}, y_i), \quad s_i = g(s_{i\text{-}1}, \boldsymbol{x}, i, y_i)$$

This slight change adds a lot of power to CRF because it provides the model with much more information that it can use for learning complex relations. Finally, the best candidate can be found by solving:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathcal{Y}^n}{\arg\max}\ \boldsymbol{w} \cdot \left[\sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{x}, i, s_{i\text{-}1}, y_i) \right] \qquad (1)$$

4 Model description: headlines as bitmaps

We model headline generation as a sequence prediction task. In this manner a news article is seen as a series of observations, where each is a possible token in the document. Furthermore, each observation can be assigned to one of two categories: in-headline, or not in-headline. Note that this approach allows a generated headline to be interpreted as a bitmap over the article's tokens.

If this set-up was used for a CRF model, the standard local feature function $\boldsymbol{f}(\boldsymbol{x}, i, y_{i\text{-}1}, y_i)$ would only be able to know whether the previous token was taken or not, which would not be very informative. For solving the problem we integrate a state sequence into the model, where a state $s \in \mathcal{S}$ holds the following information:

- The last token that was chosen as part of the headline.
- The part-of-speech tag[1] of the second-to-last word that was selected as part of the headline.

[1] We used the set of 45 tags from the Penn Treebank Tag-Set.

- The number of words already chosen to be part of the headline, which could be zero.

Therefore, the local feature function $\boldsymbol{f}(\boldsymbol{x}, i, s_{i\text{-}1}, y_i)$ will not only know about the whole text and the current token x_i, whose category y_i is to be assigned, but it will also hold information about the headline constructed so far. In our model, the objective of the local feature function is to return a vector that describes in an abstract euclidean space the outcome of placing, or not placing, token x_i in the headline, provided that the words previously chosen form the state $s_{i\text{-}1}$. We decide to make this feature vector consist of 23 signals, which only fire if the token x_i is placed in the headline (i.e., $y_i = 1$). The signals can be grouped into the following five sets:

Language-model features: they assess the grammaticality of the headline being built. The first feature is the bigram probability of the current token, given the last one placed on the headline. The second feature is the trigram probability of the PoS tag of the current token, given the tags of the two last tokens on the headline.

Keyword features: binary features that help the model detect whether the token under analysis is a salient word. The document's keywords are calculated as a preprocessing step via TF-IDF weighting, and the features fire depending on how good or bad the current token x_i is ranked with respect to the others on the text.

Dependency features: in charge of informing the model about syntactical dependencies among the tokens placed on the headline. For this end the dependency tree of all the sentences in the news article are computed as a pre-processing step[2].

Named-entity features: help the system identify named entities[3] in the text, including those that are composed of contiguous tokens.

Headline-length features: responsible for enabling the model to decide whether a headline is too short or too long. As many previous studies report, an ideal headline must have from 8 to 10 tokens (Banko et al., 2000). Thus, we include three binary features that correspond to the following conditions: (1) if the headline length so far is less than or equal to seven; (2) if the headline length so far is greater than or equal to 11; (3) if the token under analysis is assigned to the headline, which is a bias feature.

[2]We use the Stanford toolkit for computing parse trees.

[3]We only use PER, LOC, and ORG as entity annotations.

5 Decoding the model

Decoding the model involves solving equation (1) being given a weight vector $\boldsymbol{w}$, whose value stays unchanged during the process. A naive way of solving the optimization problem would be to try all possible $|\mathcal{Y}|^n$ sequence combinations, which would lead to an intractable procedure. In order to design a polynomial algorithm that finds the global optimum of the aforementioned formula, the following four observations must be made:

(1) Our model is designed so that the local feature function only fires when a token is selected for being part of a headline. Then, when evaluating an arbitrary solution $\boldsymbol{y}$, only the tokens placed on the headline must be taken into account.

(2) When applying the local feature function to a particular token x_i (assuming $y_i = 1$), the result of the function will vary only depending on the provided previous state $s_{i\text{-}1}$; all the other parameters are fixed. Moreover, a new state s_i will be generated, which in turn will include token x_i. This implies that the entire evaluation of a solution can be completely modeled as a sequence of state transitions; i.e., it becomes possible to recover a solution's bitmap from a sequence of state transitions and vice-versa.

(3) When analyzing the local feature function at any token x_i, the amount of different states $s_{i\text{-}1}$ that can be fed to the function depend solely on the tokens taken before, for which there are $2^{i\text{-}1}$ different combinations. Nevertheless, because a state only holds three pieces of information, a better upper-bound to the number of possible reachable states is equal to $i^2 \times |PoS|$, which accounts to: all possible candidates for the last token chosen before x_i, times all possible combinations of total number of tokens taken before x_i, times all possible PoS tags of the one-before-last token taken before x_i.

(4) The total amount of producible states in the whole text is equal to $\sum_{i=1}^{n} i^2 \times |PoS| = O(n^3 \times |PoS|)$. If the model is also constrained to produce headlines containing no more than H tokens, the asymptotic bound drops to $O(H \times n^2 \times |PoS|)$.

The final conclusion of these observations is as follows: since any solution can be modelled as a chain of state sequences, the global optimum can be found by generating all possible states and fetching the one that, when reached from the initial state, yields the maximum score. This task is achievable with a number of operations linearly proportional to the number of possible states, which at the same time is polynomial with respect to the number of tokens in the document. In conclusion, the model can be decoded in quadratic time. The pseudo-code in algorithm 1 gives a sketch of a $O(H \times n^2 \times |PoS|)$ bottom-up implementation.

Algorithm 1 Sketch of a bottom-up algorithm for finding the top-scoring state s^* that leads to the global optimum of our model's objective function. It iteratively computes two functions: $\pi(i, l)$, which returns the set of all reachable states that correspond to headlines having token-length l and finishing with token x_i, and $\alpha(s)$, which returns the maximum score that can be obtained by following a chain of state sequences that ends in the provided state, and starts on s_0.

//Constants.
$H \leftarrow$ Max. number of allowed tokens in headlines.
$n \leftarrow$ Number of tokens in the document.
$\boldsymbol{x} \leftarrow$ List of n tokens (document).
$\boldsymbol{w} \leftarrow$ Weight vector.
$g \leftarrow$ State transition function.
$\boldsymbol{f} \leftarrow$ Local feature function.
$s_0 \leftarrow$ Init state.
//Variables.
$\pi \leftarrow$ **new** Set<State>$[n+1][H+1](\{\})$
$\alpha \leftarrow$ **new** Float[|State|]$(-\infty)$
$s^* \leftarrow s_0$
//Base cases.
$\alpha(s_0) \leftarrow 0$
for i in $\{0, ..., n\}$ **do**
    $\pi(i, 0) \leftarrow \{s_0\}$
//Bottom-up fill of π and α.
for l in $\{1, ..., H\}$ **do**
    for i in $\{l, ..., n\}$ **do**
        for j in $\{l-1, ..., i-1\}$ **do**
            for z in $\pi(j, l-1)$ **do**
                $s \leftarrow g(z, x, i, 1)$
                $s_{\text{score}} \leftarrow \alpha(z) + \boldsymbol{w} \cdot \boldsymbol{f}(\boldsymbol{x}, i, z, 1)$
                $\pi(i, l) \leftarrow \pi(i, l) \cup \{s\}$
                $\alpha(s) \leftarrow max(\alpha(s), s_{\text{score}})$
                if $\alpha(s) > \alpha(s^*)$ **then**
                    $s^* \leftarrow s$

6 Training the model: learning what human-generated headlines look like

The global feature function $\boldsymbol{F}$ is responsible for taking a document and a bitmap, and producing a vector that describes the candidate headline in an abstract feature space. We defined the feature function so it only focuses on evaluating how a series of tokens that comprise a headline relate to each other and to the document as a whole. This implies that if $\boldsymbol{h} = \{h_1, h_2, ..., h_k\}$ is the tokenized form of any arbitrary headline consisting of k tokens, and we define vectors $\boldsymbol{a} \in \mathcal{X}^{k+n}$ and $\boldsymbol{b} \in \mathcal{Y}^{k+n}$ as:

$$\boldsymbol{a} = \{h_1, h_2, ..., h_k, x_1, x_2, ..., x_n\}$$

$$\boldsymbol{b} = \{1_1, 1_2, ..., 1_k, 0_1, 0_2, ..., 0_n\}$$

where $\boldsymbol{a}$ is the concatenation of $\boldsymbol{h}$ and $\boldsymbol{x}$, and $\boldsymbol{b}$ is a bitmap for only selecting the actual headline tokens, it follows that the feature vector that results from calling the global feature function, which we define as

$$\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{a}, \boldsymbol{b}) \tag{2}$$

is equivalent to a description of how headline $\boldsymbol{h}$ relates to document $\boldsymbol{x}$. This observation is the core of our learning algorithm, because it implies that it is possible to "insert" a human-generated headline in the text and get its description in the abstract feature space induced by $\boldsymbol{F}$. The objective of the learning process will consist in molding a weight vector $\boldsymbol{w}$, such that it makes the decoding algorithm favor headlines whose descriptions in feature space resemble the characteristics of human-generated titles.

For training the model we follow the on-line learning schemes presented by Collins (2002) and applied in studies that deal with CRF models with state sequences, such as the dependency parsing model of McDonald et al. (2005). The learning framework consists of an averaged perceptron that iteratively sifts through the training data and performs the following error-correction update at each step:

$$\boldsymbol{w}^* \leftarrow \boldsymbol{w} + \tau \times (\boldsymbol{u} - \hat{\boldsymbol{v}}), \quad \hat{\boldsymbol{v}} = \boldsymbol{F}(\boldsymbol{x}, \hat{\boldsymbol{y}})$$

where $\boldsymbol{u}$ is the vector defined in equation (2), $\hat{\boldsymbol{y}}$ is the result of solving equation (1) with the current weight

vector, and $\tau \in \mathbb{R}$ is a learning factor. We try three different values for τ, which lead to the following learning algorithms:

- **Perceptron:** $\tau = 1$
- **MIRA:** $\tau = max\left(0, \frac{1-\boldsymbol{w}\cdot(\boldsymbol{u}-\hat{\boldsymbol{v}})}{||\boldsymbol{u}-\hat{\boldsymbol{v}}||^2}\right)$
- **Forced-MIRA:** $\tau = \frac{1-\boldsymbol{w}\cdot(\boldsymbol{u}-\hat{\boldsymbol{v}})}{||\boldsymbol{u}-\hat{\boldsymbol{v}}||^2}$

The first value is a simple averaged perceptron as presented by Collins (2002), the second value is a Margin Infused Relaxed Algorithm (MIRA) as presented by Crammer and Singer (2003), and the third value is a slight variation to the MIRA update. We propose it for making the algorithm acknowledge that the objective feature vector $\boldsymbol{u}$ cannot be produced from document $\boldsymbol{x}$, and thus force an update at every step. The reason for this is that if $\boldsymbol{w} \cdot \boldsymbol{u} > \boldsymbol{w} \cdot \hat{\boldsymbol{v}}$, then MIRA sets $\tau = 0$, because an error was not made (i.e. the human-generated headline got a higher score than all of the others). Nevertheless, we observed in our experiments that this behaviour biases the process towards learning weights that exploit patterns that can occur in human-generated titles, but are almost never observed in the titles that can be generated by our model, which hinders the quality of the final headlines.

7 Automatic evaluation set-up

7.1 Evaluation metrics

For performing an automatic evaluation of the headlines generated by our system we follow the path taken by related work such as Zajic et al. (2004) and use a subset of the ROUGE metrics for comparing candidate headlines with reference ones, which have been proven to strongly correlate with human evaluations (Lin, 2003; Lin, 2004).

We decide to use as metrics ROUGE-1, ROUGE-2, and ROUGE-SU. We also propose as an experimental new metric a weighted version of ROUGE-SU, which we name ROUGE-WSU. The rationale of our proposal is that ROUGE-SU gives the same importance to all skip-bigrams extracted from a phrase no matter how far apart they are. We address the problem by weighting each shared skip-gram between phrases by the inverse of the token's average gap distance. Formally:

$$Sim(R,C) = \frac{\sum_{(a,b)\in su(R)\cap su(C)} \frac{2}{dist_R(a,b)+dist_C(a,b)}}{\sum_{(a,b)\in su(R)} \frac{1}{dist_R(a,b)}}$$

Where function $dist_H(a,b)$ returns the skip distance between tokens "a" and "b" in headline H, and $su(H)$ returns all skip-bigrams in headline H.

With the objective of having a metric capable of detecting abstract concepts in phrases and comparing headlines at a semantic level, we resort to Latent Semantic Indexing (LSI) (Deerwester et al., 1990). We use the method for extracting latent concepts from our training corpus so as to be able to represent text in an abstract latent space. We then compute the similarity of a headline with respect to a news article by calculating the cosine similarity of their vector representations in latent space.

7.2 Baselines

In order to have a point of reference for interpreting the performance of our model, we implement four baseline models. We arbitrarily decide to make all the baselines generate, if possible, nine-token-long headlines, where the last token must always be a period. This follows from the observation that good headlines must contain about eight tokens (Banko et al., 2000). The implemented baselines are the following:

Chunked first sentence: the first eight tokens from the article, plus a period at the end.

Hidden Markov Model: as proposed by Zajic et al. (2002), but adapted for producing eight-token sentences, plus an ending period.

Word Graphs: as proposed by Filippova (2010). This is a state-of-the-art multi-sentence compression algorithm. To ensure it produces headlines as output, we keep the shortest path in the graph with length equal to or greater than eight tokens. An ending period is appended if not already present. Note that the original algorithm would produce the top-k shortest paths and keep the one with best average edge weight, not caring about its length.

Keywords: the top eight keywords in the article, as ranked by TF-IDF weighting, sorted in descending order of relevance. This is not a real baseline

because it does not produce proper headlines, but it is used for naively trying to maximize the achievable value of the evaluation metrics. This is based on the assumption that keywords are the most likely tokens to occur in human-generated headlines.

7.3 Experiments and Results

We trained our model with a corpus consisting of roughly 1.3 million financial news articles fetched from the web, written in English, and published on the second half of 2012. We decided to add three important constraints to the learning algorithm which proved to yield positive empirical results:

(1) Large news articles are simplified by eliminating their most redundant or least informative sentences. For this end, the text ranking algorithm proposed by Mihalcea and Tarau (2004) is used for discriminating salient sentences in the article. Furthermore, a news article is considered large if it has more than 300 tokens, which corresponds to the average number of words per article in our training set.

(2) Because we observed that less than 2% of the headlines in the training set contained more than 15 tokens, we constraint the decoding algorithm to only generate headlines consisting of 15 or fewer tokens.

(3) We restrain the decoding algorithm from placing symbols such as commas, quotation marks, and question marks on headlines. Nonetheless, only headlines that end with a period are considered as solutions; otherwise the model tends to generate non-conclusive phrases as titles.

For automated testing purposes we use a training set consisting of roughly 12,000 previously unseen articles, which were randomly extracted from the initial dataset before training. The evaluation consisted in producing seven candidate headlines per article: one for each of the four baselines, plus one for each of the three variations of our model (each differing solely on the scheme used to learn the weight vector). Then each candidate is compared against the article's reference headline by means of the five proposed metrics.

Table 1 summarizes the obtained results of the models with respect to the ROUGE metrics. The results show that our model, when trained with our proposed forced-MIRA update, outperforms all the other baselines on all metrics, except for ROUGE-2, where all differences are statistically significant when assessed via a paired t-test ($p < 0.001$). Also, as initially intended, the keywords baseline does produce better scores than all the other methods, therefore it is considered as a naive upper-bound. It must be highlighted that all the numbers on the table are rather low, this occurs because, as noted by Zajic et al. (2002), humans tend to use a very different vocabulary and writing-style on headlines than on articles. The effect of this is that our methods and baselines are not capable of producing headlines with wordings strongly similar to human-written ones, which as a consequence makes it almost impossible to obtain high ROUGE scores.

	R-1	R-2	R-SU	R-WSU
Perceptron	0.157	0.056	0.053	0.082
MIRA	0.172	0.042	0.057	0.084
f-MIRA	**0.187**	0.054	**0.065**	**0.095**
1st sent.	0.076	0.021	0.025	0.038
HMM	0.090	0.009	0.023	0.038
Word graphs	0.174	**0.060**	0.060	0.084
Keywords	**0.313**	0.021	**0.112**	**0.148**

Table 1: Result of the evaluation of our models and baselines with respect to ROUGE metrics.

For having a more objective assessment of our proposal, we carried out a human evaluation of the headlines generated by our model when trained with the f-MIRA scheme and the word graphs approach by Filippova (2010). For this purpose, 100 articles were randomly extracted from the test set and their respective candidate headlines were generated. Then different human raters were asked to evaluate on a Likert scale, from 1 to 5, both the grammaticality and informativeness of the titles. Each article-headline pair was annotated by three different raters. The median of their ratings was chosen as a final mark. As a reference, the raters were also asked to annotate the actual human-generated headlines from the articles, although they were not informed about the provenance of the titles. We measured inter-judge agreement by means of their Intra-Class Correlation (ICC) (Cicchetti, 1994). The ICC for grammaticality was 0.51 ± 0.07, which represents fair agreement, and the ICC for informativeness was 0.63 ± 0.05, which represents substantial agreement.

Table 2 contains the results of the models with

	H. Len.	LSI	Gram.	Inf.
Perceptron	10.096	0.446	–	–
MIRA	13.045	0.463	–	–
f-MIRA	11.737	**0.491**	3.45	**2.94**
1stsent.	8.932	0.224	–	–
HMM	9.000	0.172	–	–
Word graphs	10.973	0.480	**3.69**	2.32
Keywords	9.000	**0.701**	–	–
Reference	11.898	**0.555**	**4.49**	**4.14**

Table 2: Result of the evaluation of our models and baselines with LSI document similarity, grammaticality and informativeness as assessed by human raters, and average headline length.

respect to the LSI document similarity metric, and the human evaluations for grammaticality and informativeness. For exploratory purposes the table also contains the average length for the generated headlines of each of the models (which also counts the imposed final period). The results in this table are satisfying: with respect to LSI document similarity, our model outperforms all of the baselines and its value is close to the one achieved by human-generated headlines. On the other hand, the human evaluations are middling: the word-graphs method produces more readable headlines, but our model proves to be more informative because it does better work at detecting abstract word relationships in the text. All differences in this table are statistically significant when computed as paired t-tests ($p < 0.001$).

It is worth noting that the informativeness of human-generated headlines did not get a high score. The reason for this is the fact that editors tend to produce rather sensationalist or partially informative titles so as to attract the attention of readers and engage them to read the whole article; human raters penalized the relevance of such headlines, which was reflected on this final score.

Finally, table 3 contains the mutual correlation between automated and manual metrics. The first thing to note is that none of the used metrics proved to be good for assessing grammaticality of headlines. It is also worth noting that our proposed metric ROUGE-WSU performs as well as the other ROUGE metrics, and that the proposed LSI document similarity does not prove to be as strong a metric as the others.

	R-1	R-2	R-SU	R-WSU	LSI-DS
Gram.	-0.130	-0.084	-0.131	-0.132	-0.015
Inf.	0.561	0.535	0.557	0.542	0.370

Table 3: Spearman correlation between human-assessed metrics and automatic ones.

8 Conclusions and Discussion

In this study we proposed a CRF model with state transitions. The model tries to learn how humans title their articles. The learning is performed by means of a mapping function that, given a document, translates headlines to an abstract feature-rich space, where the characteristics that distinguish human-generated titles can be discriminated. This abstraction allows our model to be trained with any monolingual corpus of news articles because it does not impose conditions on the provenance of stories' headlines –i.e, our model maps reference headlines to a feature space and only learns what abstract properties characterize them. Furthermore, our model allows defining the task of finding the best possible producible headline as a discrete optimization problem. By doing this each candidate headline is modelled as a path in a graph of state sequences, thus allowing the best-scoring path to be found in polynomial time by means of dynamic programming. Our results, obtained through reliable automatic and human-assessed evaluations, provide a proof of concept for the soundness of our model and its capabilities. Additionally, we propose a new evaluation metric, ROUGE-WSU, which, as shown in table 3, correlates as good as traditional ROUGE metrics with human evaluations for informativeness of headlines.

The further work we envisage for augmenting our research can be grouped in the following areas:

- Exploring more advanced features that manage to detect abstract semantic relationships or discourse flows in the compressed article.
- Complementing our system with a separate translation model capable of transforming to "Headlinese" the titles generated with the language used in the bodies of articles.
- Attempting to achieve a more objective evaluation of our generated headlines, through the use of semantic-level measures.

Acknowledgments

We thank Jordi Atserias, Horacio Saggion, Horacio Rodríguez, and Xavier Carreras, who helped and supported us during the development of this study.

References

[Banko et al.2000] Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*.

[Berger and Mittal2000] B. Berger and N. Mittal. 2000. Discourse segmentation in aid of document summarization. *Proceedings of the Hawaii International Conference on System Sciences, Minitrack on Digital Documents Understanding*.

[Borko and Bernier1987] H. Borko and C. Bernier. 1987. Abstracting concepts and methods. *New York: Academic Press*.

[Buyukkokten et al.2001] Orkut Buyukkokten, Hector Garcia-Molina, and Andreas Paepcke. 2001. Seeing the whole in parts: text summarization for web browsing on handheld devices. *Proceedings of the 10th international conference on World Wide Web. ACM*.

[Cicchetti1994] Domenic V. Cicchetti. 1994. Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological assessment 6.4*.

[Collins2002] Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*.

[Crammer and Singer2003] Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research 3*.

[Das and Martins2007] Dipanjan Das and André FT Martins. 2007. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU 4*, pages 192–195.

[De Kok2008] D. De Kok. 2008. Headline generation for dutch newspaper articles through transformation-based learning. *Master Thesis*.

[Deerwester et al.1990] Scott C. Deerwester et al. 1990. Indexing by latent semantic analysis. *JASIS 41.6*.

[Dorr et al.2003] Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: a parse-and-trim approach to headline generation. *Proceedings of the HLT-NAACL 03 on Text summarization workshop*, pages 1–8.

[Filippova2010] Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. *Proceedings of the 23rd International Conference on Computational Linguistics*.

[Gattani2007] Akshay Kishore Gattani. 2007. Automated natural language headline generation using discriminative machine learning models. *Diss. Simon Fraser University*.

[Hajime et al.2013] Morita Hajime et al. 2013. Subtree extractive summarization via submodular maximization. *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics*.

[Jin and Hauptmann2001] Rong Jin and Alexander G. Hauptmann. 2001. Title generation using a training corpus. *CICLing '01: Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing*, pages 208–215.

[Knight and Marcu2001] Kevin Knight and Daniel Marcu. 2001. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence 139.1*, pages 91–107.

[Kupiec1992] Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language 6.3*.

[Lafferty et al.2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[Lin2003] C.Y. Lin. 2003. Cross-domain study of n-gram co-occurrence metrics. *Proceedings of the Workshop on Machine Translation Evaluation, New Orleans, USA*.

[Lin2004] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

[Linke-Ellis1999] N. Linke-Ellis. 1999. Closed captioning in america: Looking beyond compliance. *Proceedings of the TAO Workshop on TV Closed Captions for the Hearing Impaired People, Tokyo, Japan*, pages 43–59.

[Mani and Maybury1999] Inderjeet Mani and Mark T. Maybury Maybury. 1999. *Advances in automatic text summarization*, volume 293. Cambridge: MIT press.

[Martins and Smith2009] André F.T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. *NAACL-HLT Workshop on Integer Linear Programming for NLP, Boulder, USA*.

[McCallum and Li2003] Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with

conditional random fields, feature induction and web-enhanced lexicons. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*.

[McCallum et al.2000] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. *ICML*.

[McDonald et al.2005] Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

[McDonald2006] Ryan T. McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. *EACL*.

[Mihalcea and Tarau2004] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. *Association for Computational Linguistics*.

[Mohri et al.2002] Mehryar Mohri, Fernando Pereira, and Michael Richard. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language 16.1*.

[Nenkova and McKeown2012] Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *Mining Text Data. Springer US*, pages 43–76.

[Nomoto2007] Tadashi Nomoto. 2007. Discriminative sentence compression with conditional random fields. *Information processing and management 43.6*.

[Rabiner and Juang1986] Lawrence Rabiner and Biing-Hwang Juang. 1986. An introduction to hidden markov models. *ASSP Magazine, IEEE 3.1*, pages 4–16.

[Reynar and Ratnaparkhi1997] Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. *Proceedings of the fifth conference on Applied natural language processing*.

[Shen et al.2007] Dou Shen et al. 2007. Document summarization using conditional random fields. *IJCAI. Vol. 7*.

[Unno et al.2006] Yuya Unno et al. 2006. Trimming cfg parse trees for sentence compression using machine learning approaches. *Proceedings of the COLING/ACL on Main conference poster sessions*.

[Zajic et al.2002] David Zajic, Bonnie Dorr, and Richard Schwartz. 2002. Automatic headline generation for newspaper stories. *Workshop on Automatic Summarization*.

[Zajic et al.2004] David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*.