HEADY: News headline abstraction through event pattern clustering

Enrique Alfonseca
Google Inc.
ealfonseca@google.com

Daniele Pighin
Google Inc.
biondo@google.com

Guillermo Garrido*
NLP & IR Group at UNED
ggarrido@lsi.uned.es

Abstract

This paper presents HEADY: a novel, *abstractive* approach for headline generation from news collections. From a web-scale corpus of English news, we mine syntactic patterns that a Noisy-OR model generalizes into event descriptions. At inference time, we query the model with the patterns observed in an unseen news collection, identify the event that better captures the gist of the collection and retrieve the most appropriate pattern to generate a headline. HEADY improves over a state-of-the-art open-domain title abstraction method, bridging half of the gap that separates it from extractive methods using human-generated titles in manual evaluations, and performs comparably to human-generated headlines as evaluated with ROUGE.

1 Introduction

Motivation. News events are rarely reported only in one way, from a single point of view. Different news agencies will interpret the event in different ways; various countries or locations may highlight different aspects of it depending on how they are affected; and opinions and in-depth analyses will be written after the fact.

The variety of contents and styles is both an opportunity and a challenge. On the positive side, we have the same events described in different ways; this redundancy is useful for summarization, as the information content reported by the majority of news sources most likely represents the central part of the event. On the other hand, variability and subjectivity can be difficult to isolate. For some applications it is important to understand, given a collection of related news articles and reports, how to formulate in an objective way what has happened.

- Carmelo and La La Party It Up with Kim and Ciara
- La La Vazquez and Carmelo Anthony: Wedding Day Bliss
- Carmelo Anthony, actress LaLa Vazquez wed in NYC
- Stylist to the Stars
- LaLa, Carmelo Set Off Celebrity Wedding Weekend
- Ciara rocks a sexy Versace Spring 2010 mini to LaLa Vasquez and Carmelo Anthony's wedding (photos)
- Lala Vasquez on her wedding dress, cake, reality tv show and fiancé, Carmelo Anthony (video)
- VAZQUEZ MARRIES SPORTS STAR ANTHONY
- Lebron Returns To NYC For Carmelo's Wedding
- Carmelo Anthony's stylist dishes on the wedding
- Paul pitching another Big Three with "Melo in NYC"
- Carmelo Anthony and La La Vazquez Get Married at Star-Studded Wedding Ceremony

Table 1: Headlines observed for a news collection reporting the same wedding event.

As a motivating example, Table 1 shows the different headlines observed in news reporting the wedding between basketball player Carmelo Anthony and actress LaLa Vazquez. As can be seen, there is a wide variety of ways to report the same event, including different points of view, highlighted aspects, and opinionated statements on the part of the reporter. When presenting this event to a user in a news-based information retrieval or recommendation system, different event descriptions may be more appropriate. For example, a user may only be interested in objective, informative summaries without any interpretation on the part of the reporter. In this case, *Carmelo Anthony, ac-*

*Work done during an internship at Google Zurich.

tress LaLa Vazquez wed in NYC would be a good choice.

Goal. Our final goal in this research is to build a headline generation system that, given a news collection, is able to describe it with the most compact, objective and informative headline. In particular, we want the system to be able to:

- Generate headlines in an open-domain, unsupervised way, so that it does not need to rely on training data which is expensive to produce.
- Generalize across synonymous expressions that refer to the same event.
- Do so in an abstractive fashion, to enforce novelty, objectivity and generality.

In order to advance towards this goal, this paper explores the following questions:

- What is a good way of using syntactic patterns to represent events for generating headlines?
- Can we have satisfactory readability with an open-domain abstractive approach, not relying on training data nor on manually predefined generation templates?
- How far can we get in terms of informativeness, compared to the human-produced headlines, i.e., extractive approaches?

Contributions. In this paper we present HEADY, which is at the same time a novel system for abstractive headline generation, and a smooth clustering of patterns describing the same events. HEADY is fully open-domain and can scale to web-sized data. By learning to generalize events across the boundaries of a single news story or news collection, HEADY produces compact and effective headlines that objectively convey the relevant information.

When compared to a state-of-the-art open-domain headline abstraction system (Filippova, 2010), the new headlines are statistically significantly better both in terms of readability and informativeness. Also, automatic evaluations using ROUGE, having objective headlines for the news as references, show that the abstractive headlines are on par with human-produced headlines.

2 Related work

Headline generation and summarization. Most headline generation work in the past has focused on the problem of single-document summarization: given the main passage of a single news article, generate a very short summary of the article. From early in the field, it was pointed out that a purely extractive approach is not good enough to generate headlines from the body text (Banko et al., 2000). Sometimes the most important information is distributed across several sentences in the document. More importantly, quite often, the single sentence selected as the most informative for the news collection is already longer than the desired headline size. For this reason, most early headline generation work focused on either extracting and reordering n-grams from the document to be summarized (Banko et al., 2000), or extracting one or two informative sentences from the document and performing linguistically-motivated transformations to them in order to reduce the summary length (Dorr et al., 2003). The first approach is not guaranteed to produce grammatical headlines, whereas the second approach is tightly tied to the actual wording found in the document. Single-document headline generation was also explored at the Document Understanding Conferences between 2002 and 2004[1].

In later years, there has been more interest in problems such as sentence compression (Galley and McKeown, 2007; Clarke and Lapata, 2008; Cohn and Lapata, 2009; Napoles et al., 2011; Berg-Kirkpatrick et al., 2011), text simplification (Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011) and sentence fusion (Barzilay and McKeown, 2005; **?**; Filippova and Strube, 2008; Elsner and Santhanam, 2011). All of them have direct applications for headline generation, as it can be construed as selecting one or a few sentences from the original document(s), and then reducing them to the target title size. For example, **?**) generate novel utterances by combining Prim's maximum-spanning-tree algorithm with an n-gram language model to enforce fluency. Unlike HEADY, the method by Wan and colleagues is an extractive method that can summarize single documents into a sentence, as opposed to generating a sentence that can stand for a whole collection of news. Filippova (2010) reports a system

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

that is very close to our settings: the input is a collection of related news articles, and the system generates a headline that describes the main event. This system uses sentence compression techniques and benefits from the redundancy in the collection. One difference with respect to HEADY is that it does not use any syntactic information aside from part-of-speech tags, and it does not require a training step. We have used this approach as a baseline for comparison.

There are not many fully abstractive systems for news summarization. The few that exist, such as the work by Genest and Lapalme (2012), rely on manually written generation templates. In contrast, HEADY automatically learns the templates or headline patterns automatically, which allows it to work in open-domain settings without relying on supervision or manual annotations.

Open-domain pattern learning. Pattern learning for relation extraction is an active area of research that is very related to our problem of event pattern learning for headline generation. TextRunner (Yates et al., 2007), ReVerb (Fader et al., 2011) and NELL (Carlson et al., 2010; Mohamed et al., 2011) are some examples of open-domain systems that learn surface patterns that express relations between pairs of entities. PATTY (Nakashole et al., 2012) generalizes the patterns to also include syntactic information and ontological (class membership) constraints. Our patterns are more similar to the ones used by PATTY, which also produces clusters of synonymous patterns. The main differences are that (a) HEADY is not limited to consider patterns expressing relations between pairs of entities; (b) we identify synonym patterns using a probabilistic, Bayesian approach that takes advantage of the multiplicity of news sources reporting the same events. **?**) present an unsupervised method for learning narrative schemas from news, i.e., coherent sets of events that involve specific entity types (semantic roles). Similarly to them, we move from the assumptions that 1) utterances involving the same entity types within the same document (in our case, a collection of related documents) are likely describing aspects of the same event, and 2) meaningful representations of the underlying events can be learned by clustering these utterances in a principled way.

Noisy-OR networks. Noisy-OR Bayesian networks (Pearl, 1988) have been applied in the past to a wide class of large-scale probabilistic inference problems, from the medical domain (Middleton et al., 1991; Jaakkola and Jordan, 1999; Onisko et al., 2001), to synthetic image-decomposition and co-citation data analysis (Šingliar and Hauskrecht, 2006). By assuming independence between the causes of the hidden variables, noisy-OR models tend to be reliable (Friedman and Goldszmidt, 1996) as they require a relatively small number of parameters to be estimated (linear with the size of the network).

3 Headline generation

In this section, we describe the HEADY system for news headline abstraction. Our approach takes as input, for training, a corpus of news articles organized in news collections. Once the model is trained, it can generate headlines for new collections. An outline of HEADY's main components follows (details of each component are provided in Sections 3.1, 3.2 and 3.3):

Pattern extraction. Identify, in each of the news collections, syntactic patterns connecting k entities, for $k \geq 1$. These will be the candidate patterns expressing events.

Training. Train a Noisy-OR Bayesian network on the co-occurrence of syntactic patterns. Each pattern extracted in the previous step is added as an observed variable, and latent variables are used to represent the hidden events that generate patterns. An additional *noise* variable links to every terminal node, allowing every terminal to be generated by language background (noise) instead of by an actual event.

Inference. Generate a headline from an unseen news collection. First, patterns are extracted using the pattern extraction procedure mentioned above. Given the patterns, the posterior probability of the hidden *event* variables is estimated. Then, from the activated hidden events, the likelihood of every pattern can be estimated, even if they do not appear in the collection. The single pattern with the maximum probability is selected to generate a new headline from it. Being the product of extra-news collection generalization, the retrieved pattern is more likely to be objective and informative than patterns directly observed in the news collection.

Algorithm 1 COLLECTIONTOPATTERNS$_\Psi(\mathcal{N})$: $\mathcal{N}$ is a repository of news collections, Ψ is a set of parameters controlling the extraction process.

$\mathcal{R} \leftarrow \{\}$
for all $N \in \mathcal{N}$ **do**
  PREPROCESSDATA(N)
  $E \leftarrow$ GETRELEVANTENTITIES(N')
  for all $E_i \leftarrow$ COMBINATIONS$_\Psi(E)$ **do**
    for all $\boldsymbol{n} \in N$ **do**
      $\mathcal{P} \leftarrow$ EXTRACTPATTERNS$_\Psi(\boldsymbol{n}, E_i)$
      $\mathcal{R}\{N, E_i\} \leftarrow \mathcal{R}\{N, E_i\} \cup P$
return $\mathcal{R}$

3.1 Pattern extraction

In this section we detail the process for obtaining the event patterns that constitute the building blocks of learning and inference.

Patterns are extracted from a large repository $\mathcal{N}$ of news collections $N_1, \ldots, N_{|\mathcal{N}|}$. Each news collection $N = \{\boldsymbol{n}_i\}$ is an unordered collection of related news, each of which can be seen as an ordered sequence of sentences, i.e.: $\boldsymbol{n} = [s_0, \ldots s_{|n|}]$.

Algorithm 1 presents a high-level view of the pattern extraction process. The different steps are described below:

PREPROCESSDATA: We start by preprocessing all the news in the news collections with a standard NLP pipeline: tokenization and sentence boundary detection (Gillick, 2009), part-of-speech tagging, dependency parsing (Nivre, 2006), coreference resolution (Haghighi and Klein, 2009) and entity linking based on Wikipedia and Freebase. Using the Freebase dataset, each entity is annotated with all its Freebase types (class labels). In the end, for each entity mentioned in the document we have a unique identifier, a list with all its mentions in the document and a list of class labels from Freebase.

As a result of this process, we obtain for each sentence in the corpus a representation as exemplified in Figure 1 (1). In this example, the mentions of three distinct entities have been identified, i.e., $e_1, \ldots, e_3$. In the Freebase list of types (class labels), e_1 is a *person* and a *celebrity*, and e_3 is a *state* and a *location*.

GETRELEVANTENTITIES: For each news collection N we collect the set E of the entities mentioned most often within the collection. Next, we generate the set COMBINATIONS$_\Psi(E)$ consisting of non-empty subsets of E, without repeated entities. The number of entities to consider in each collection, and the maximum size for the subsets of entities to consider are meta-parameters embedded in Ψ.[2]

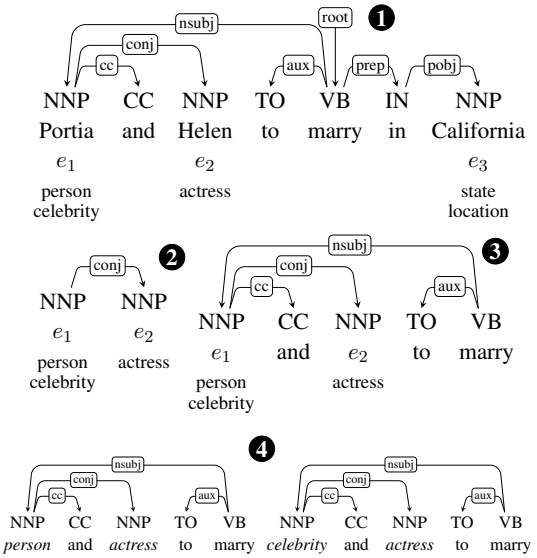

Figure 1: Pattern extraction process from an annotated dependency parse. (1): an MST is extracted from the entity pair e_1, e_2 (2); nodes are heuristically added to the MST to enforce grammaticality (3); entity types are recombined to generate the final patterns (4).

EXTRACTPATTERNS: For each subset of relevant entities E_i, event patterns are mined from the articles in the news collection. The process by which patterns are extracted from a news is explained in Algorithm 2 and exemplified graphically in Figure 1 (2–4).

GETMENTIONNODES: Using the dependency parse T for a sentence s, we first identify the set of nodes M_i that mention the entities in E_i. If T does not contain exactly one mention of each target entity in E_i, then the sentence is ignored. Otherwise, we obtain the minimum spanning tree for the nodeset P_i, i.e., the shortest path in the dependency tree connecting all the nodes in M_i (Figure 1, 2). P_i is the set of nodes around which the patterns will be constructed.

APPLYHEURISTICS: With very high probability, the MST P_i that we obtain does not constitute a grammatical or useful extrapolation of the original sentence s. For example, the MST for the en-

[2] As our objective is to generate very short titles (under 10 words), we only consider combinations of up to three elements of E.

Algorithm 2 EXTRACTPATTERNS$_\Psi(\boldsymbol{n}, E_i)$: $\boldsymbol{n}$ is the list of sentences in a news article. Sentences are POS-tagged, dependency parsed and annotated with respect to a set of entities $E \supseteq E_i$

$\mathcal{P} \leftarrow \emptyset$
for all $s \in \boldsymbol{n}[0:2)$ **do**
  $T \leftarrow$ DEPPARSE(s)
  $M_i \leftarrow$ GETMENTIONNODES(t, E_i)
  if $\exists e \in E_i, count(e, M_i) \neq 1$ **then continue**
  $P_i \leftarrow$ GETMINIMUMSPANNINGTREE$_\Psi(M_i)$
  APPLYHEURISTICS$_\Psi(P_i)$ **or continue**
  $\mathcal{P} \leftarrow \mathcal{P} \cup$ COMBINEENTITYTYPES$_\Psi(P_i)$
return $\mathcal{P}$

tity pair $\langle e_1, e_2 \rangle$ in the example does not provide a good description of the event as it is neither adequate nor fluent. For this reason, we apply a set of post-processing heuristic transformations that aim at including a minimal set of meaningful nodes. These include making sure that both the root of the clause and its subject appear in the extracted pattern, and that conjunctions between entities should not be dropped (Figure 1, 3).

COMBINEENTITYTYPES: Finally, a distinct pattern is generated from each possible combination of entity type assignments for the participating entities. (Figure 1, 4).

It is important to note that both at training and test time, for pattern extraction we only consider the title and the first sentence of the article body. The reason is that we want to limit ourselves, in each news collection, to the most relevant event reported in the collection, which appears most of the times in these two sentences. Unlike titles, first sentences do not extensively use puns or rhetorics as they tend to be grammatical and informative rather than catchy.

The patterns mined from the same news collection and for the same set of entities are grouped together, and constitute the building blocks of the clustering algorithm which is described below.

3.2 Training

The extracted patterns are used to learn a Noisy-OR (Pearl, 1988) model by estimating the probability that each (observed) pattern activates one or many (hidden) *events*. Figure 2 represents the two levels: the hidden event variables at the top, and the observed pattern variables at the bottom. An additional *noise* variable links to every terminal node, allowing all terminals to be generated by language background (noise) instead of by an actual event. The associations between latent events and observed patterns are modeled by noisy-OR gates.

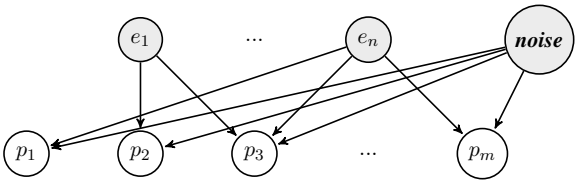

Figure 2: Probabilistic model. The associations between latent event variables and observed pattern variables are modeled by noisy-OR gates. Events are assumed to be marginally independent, and patterns conditionally independent given the events.

In this model, the conditional probability of a hidden event e_i given a configuration of observed patterns $\boldsymbol{p} \in \{0,1\}^{|\mathcal{P}|}$ is calculated as:

$$P(e_i = 0 \mid \boldsymbol{p}) = (1 - q_{i0}) \prod_{j \in \pi_j} (1 - q_{ij})^{p_j}$$
$$= \exp\left(-\theta_{i0} - \sum_{j \in \pi_i} \theta_{ij} p_j\right),$$

where π_i is the set of active events (i.e., $\pi_i = \cup_j \{p_j\} \mid p_j = 1$), and $q_{ij} = P(e_i = 1 \mid p_j = 1)$ is the estimated probability that the observed pattern p_i can, in isolation, activate the event e. The term q_{i0} is the so-called "noise" term of the model, and it accounts for the fact that an observed event e_i might be activated by some pattern that has never been observed (Jaakkola and Jordan, 1999).

In Algorithm 1, at the end of the process we group in $\mathcal{R}[N, E_i]$ all the patterns extracted from the same news collection N and entity sub-set E_i. These groups represent rough clusters of patterns, that we can use to bootstrap the optimization of the model parameters $\theta_{ij} = -\log(1 - q_{ij})$. We initiate the training process by randomly selecting 100,000 of these groups, and optimize the weights of the model through 40 EM (Dempster et al., 1977) iterations.

3.3 Inference (generation of new headlines)

Given an unseen news collection N, the inference component of HEADY generates a single headline that captures the main event reported by the news in N. In order to do so, we first need to select a

single event-pattern p^* that is especially relevant for N. Having selected p^*, in order to generate a headline it is sufficient to replace the entity placeholders in p^* with the surface forms observed in N.

To identify p^*, we start from the assumption that the most descriptive event encoded by N must describe an important situation in which some subset of the relevant entities E in N are involved.

The basic inference algorithm is a two-step random walk in the Bayesian network. Given a set of entities E and sentences $\boldsymbol{n}$, EXTRACTPATTERNS$_\Psi(\boldsymbol{n}, E)$ collects patterns involving those entities. By normalizing the frequency of the extracted patterns, we get a probability distribution over the observed variables in the network. A two-step random walk traversing to the latent event nodes and back to the pattern nodes allows us to generalize across events. We call this algorithm INFERENCE($\boldsymbol{n}$, E).

In order to decide which is the most relevant set of events that should appear in the headline, we use the following procedure:

1. Given the set of entities E mentioned in the news collection, we consider each entity subset $E_i \subseteq E$ including up to three entities[3]. For each E_i, we run INFERENCE($\boldsymbol{n}$, E_i), which computes a distribution $\boldsymbol{w}_i$ over patterns involving the entities in E_i.

2. We invoke again INFERENCE, now using at the same time all the patterns extracted for every subset of $E_i \subseteq E$. This computes a probability distribution $\boldsymbol{w}$ over all patterns involving any admissible subset of the entities mentioned in the collection.

3. Third, we select the entity-specific distribution that approximates better the overall distribution

$$\boldsymbol{w}^* = \arg\max_i \cos(\boldsymbol{w}, \boldsymbol{w}_i)$$

   We assume that the corresponding set of entities E_i are the most central entities in the collection and therefore any headline should make sure to mention them all.

4. Finally, we select the pattern with the highest weight in $\boldsymbol{w}^*$ as the pattern that better captures the main event reported in the news collection:

$$p^* = p_j \mid w_j = \arg\max_j w_j^*$$

The headline is then produced from p^*, replacing placeholders with the entities in the document from which the pattern was extracted.

While in many cases information about entity types would be sufficient to decide about the order of the entities in the generated sentences (e.g., "[person] married in [location]" for the entity set $\{e_a =$ "Mr. Brown", $e_b =$ "Los Angeles"$\}$), in other cases class assignment can be ambiguous (e.g., "[person] killed [person]" for $\{e_a =$ "Mr. A", $e_b =$ "Mr. B"$\}$). To handle these cases, when extracting patterns for an entity set $\{e_a, e_b\}$, we keep track of the alphabetical ordering of the entities, e.g., from a news collection about "Mr. B" killing "Mr. A" we would produce patterns such as "[person:2] killed [person:1]" or "[person:1] was killed by [person:2]" since $e_a =$ "Mr. A" $<$ $e_b =$ "Mr. B". At inference time, when we query the model with such patterns we can only activate events whose assignments are compatible with the entities observed in the text, making the replacement straightforward and unambiguous.

4 Experiment settings

In our method we use patterns that are fully lexicalized (with the exception of entity placeholders) and enriched with syntactic data. Under these circumstances, the Noisy-OR can effectively generalize and learn meaningful clusters only if provided with large amounts of data. To our best knowledge, available data sets for headline generation are not large enough to support this kind of inference.

For this reason, we rely on a corpus of news crawled from the web between 2008 and 2012 which have been clustered based on closeness in time and cosine similarity, using the vector-space model and tf.idf weights. News collections with less than 5 documents are discarded[4], and those

[3] As we noted before, we impose this limitation to keep the generated headlines relatively short and to limit data sparsity issues.

[4] There is a very long tail of singleton articles, which do not offer useful examples of lexical or syntactic variation, and many very small collections that tend to be especially noisy, hence the decision to consider only collections with at least 5 documents.

larger than 50 documents are capped, by randomly picking 50 documents from the collection[5]. The total number of news collections after clustering is 1.7 million. From this set, we have set aside a few hundred collections that will remain unseen until the final evaluation.

As we have no development set, we have done no tuning of the parameters for pattern extraction nor for the Bayesian network training (100,000 latent variables to represent the different events, 40 EM iterations, as mentioned in Section 3.2). The EM iterations on the noisy-OR were distributed across 30 machines with 16 GB of memory each.

4.1 Systems used

One of the questions we wanted to answer in this research was whether it was possible to obtain the same quality with automatically abstracted headlines as with human-generated headlines. For every news collection we have as many human-generated headlines as documents. To decide which human-generated headline should be used in this comparison, we used three different methods that pick one of the collection headlines:

- **Latest headline**: selects the headline from the latest document in the collection. Intuitively this should be the most relevant one for news about sport matches and competitions, where the earlier headlines offer previews and predictions, and the later headlines report who won and the final scores.

- **Most frequent headline**: some headlines are repeated across the collection, and this method chooses the most frequent one. If there are several with the same frequency, one is taken at random[6].

- **TopicSum**: we use TopicSum (Haghighi and Vanderwende, 2009), a 3-layer hierarchical topic model, to infer the language model that is most central for the collection. The news title that has the smallest Kullback-Leibler divergence with respect the collection language model is the one chosen.

[5]Even though we did not run any experiment to find an optimal value for this parameter, 50 documents seems like a reasonable choice to avoid redundancy while allowing for considerable lexical and syntactic variation.

[6]The most frequent headline only has a tie in 6 collections in the whole test set. In 5 cases two headlines are tied at frequencies around 4, and in one case three headlines are tied at frequency 2. All six are large collections with 50 news articles, so this baseline is significantly different from a random baseline.

	R-1	R-2	R-SU4
HEADY	0.3565	0.1903	0.1966
Most frequent pattern	0.3560	0.1864	0.1959
TopicSum	0.3594	0.1821	0.1935
MSC	0.3470	0.1765	0.1855
Most frequent headline	0.3177	0.1401	0.1668
Latest headline	0.2814	0.1191	0.1425

Table 2: Results from the automatic evaluation, sorted according to the ROUGE-2 and ROUGE-SU4 scores.

A headline generation system that addresses the same application as ours is (Filippova, 2010), which generates a graph from the collection sentences and selects the shortest path between the begin and the end node traversing words in the same order in which they were found in the original documents. We have used this system, called Multi-Sentence Compression (**MSC**), for comparisons.

Finally, in order to understand whether the noisy-OR Bayesian network is useful for generalizing across patterns into *latent events*, we added a baseline that extracts all patterns from the test collection following the same COLLECTIONTOPATTERNS algorithm (including the application of the linguistically motivated heuristics), and then produces a headline straightaway from the most frequent pattern extracted. In other words, the only difference with respect to HEADY is that in this case no generalization through the Noisy-OR network is carried out, and that headlines are generated from patterns directly observed in the test news collections. We call this system **Most frequent pattern**.

4.2 Annotation activities

In order to evaluate HEADY's performance, we carried out two annotation activities.

First, from the set of collections that we had set aside at the beginning, we randomly chose 50 collections for which all the systems could generate an output, and we asked raters to manually write titles for them. As this is not a simple task to be crowdsourced, for this evaluation we relied on eight trained raters. We collected between four and five reference titles for each of the fifty news collections, to be used to compare the headline

	Readability	Informativeness
TopicSum	**4.86**	**4.63**
Most freq. headline	†‡4.61	$^{\dagger\ddagger\diamond}$4.43
Latest headline	†‡4.55	† 4.00
HEADY	† 4.28	† 3.75
Most freq. pattern	† 3.95	† 3.82
MSC	3.00	3.05

Table 3: Results from the manual evaluation. At 95% confidence, TopicSum is significantly better than all others for readability, and only indistinguishable from the most frequent pattern for informativeness. For the rest, $^{\diamond}$ means being significantly better than HEADY, ‡ than the most frequent pattern, and † than MSC.

generation methods using automatic summarization metrics.

Then, we took the output of the systems for the 50 test collections and asked human raters to evaluate the headlines:

1. Raters were shown one headline and asked to rate it in terms of **readability** on a 5-point Likert scale. In the instructions, the raters were provided with examples of ungrammatical and grammatical titles to guide them in this annotation.

2. After the previous rating is done, raters were shown a selection of five documents from the collection, and they were asked to judge the **informativeness** of the previous headline for the news in the collection, again on a 5-point Likert scale.

This second annotation was carried out by independent raters in a crowd-sourcing setting. The raters did not have any involvement with the inception of the model or the writing of the paper. They did not know that the headlines they were rating were generated according to different methods. We measured inter-judge agreement on the Likert-scale annotations using their Intra-Class Correlation (ICC) (Cicchetti, 1994). The ICC for readability is 0.76 (0.95 confidence interval [0.71, 0.80]), and for informativeness it is 0.67 (0.95 confidence interval [0.60, 0.73]). This means strong agreement for readability, and moderate agreement for informativeness.

5 Results

The COLLECTIONTOPATTERNS algorithm was run on the training set, producing a 230 million event patterns. Patterns that were obtained from the same collection and involving the same entities were grouped together, for a total of 1.7 million pattern collections. The pattern groups are used to bootstrap the Noisy-OR model training. Training the HEADY model that we used for the evaluation took around six hours on 30 cores.

Table 2 shows the results of the comparison of the headline generation systems using ROUGE (R-1, R-2 and R-SU4) (Lin, 2004) with the collected references. According to Owczarzak et al. (2012), ROUGE is still a competitive metric that correlates well with human judgements for ranking summarizers. The significance tests for ROUGE are performed using bootstrap resampling and a graphical significance test (Minka, 2002). The human annotators that created the references for this evaluation were explicitly instructed to write objective titles, which is the kind of headlines that the abstractive systems aim at generating. It is common to see real headlines that are catchy, joking, or with a double meaning, and therefore they use a different vocabulary than objective titles that simply mention what happened. TopicSum sometimes selects objective titles amongst the human-made titles and that is why it also scores very well with the ROUGE scores. But the other two criteria for choosing human-made headlines select non-objective titles much more often, and this lowers their performance when measured with ROUGE with respect to the objective references.

Table 3 lists the results of the manual evaluation of readability and informativeness of the generated headlines. The first result that we can see is the difference in the rankings between the two evaluations. Part of this difference might be due to the fact that ROUGE is not as good for discriminating between human-made and automatic summaries. In fact, in the DUC competitions, the gap between human summaries and automatic summaries was also more apparent in the manual evaluations than using ROUGE. Another part of the observed difference may be due to the design of the evaluation. The manual evaluation is asking raters to judge whether real, human-written titles that were actually used for those news are grammatical and informative. As could be expected, as these are published titles, the real titles score very good on the manual evaluation.

Some other interesting results are:

Model	Generated title
TopicSum	Modern Family's Eric Stonestreet laughs off Charlize Theron rumours
MSC	Modern Family star Eric Stonestreet is dating Charlize Theron.
Latest headline	Eric laughs off Theron dating rumours
Frequent pattern	Eric Stonestreet jokes about Charlize relationship
Frequent headline	Charlize Theron dating Modern Family star
HEADY	Eric Stonestreet not dating Charlize Theron
TopicSum	McFadzean rescues point for Crawley Town
MSC	Crawley side challenging for a point against Oldham Athletic.
Latest headline	Reds midfielder victim of racist tweet
Frequent pattern	Kyle McFadzean fired a equaliser Crawley were made
Frequent headline	Latics halt Crawley charge
HEADY	Kyle McFadzean rescues point for Crawley Town F.C.
TopicSum	UCI to strip Lance Armstrong of his 7 Tour titles
MSC	The international cycling union said today.
Latest headline	Letters: elderly drivers and Lance Armstrong
Frequent pattern	Lance Armstrong stripped of Tour de France titles
Frequent headline	Today in the news: third debate is tonight
HEADY	Lance Armstrong was stripped of Tour de France titles

Table 4: A comparison of the titles generated by the different models for three news collections.

- Amongst the automatic systems, HEADY performed better than MSC, with statistical significance at 95% for all the metrics. Headlines based on the most frequent patterns were better than MSC for all metrics but ROUGE-2.

- The most frequent pattern baseline and HEADY have comparable performance across all the metrics (not statistically significantly different), although HEADY has slightly better scores for all metrics except for informativeness.

While we do not take any step to explicitly model stylistic variation, estimating the weights of the Noisy-OR network turns out to be a very effective way of filtering out sensational wording to the advantage of plainer, more objective style. This may not clearly emerge from the evaluation, as we did not explicitly ask the raters to annotate the items based on their objectivity, but a manual inspection of the clusters suggests that the generalization is working in the right direction.

Table 4 presents a selection of outputs produced by the six models for three different news collections. The first example shows a news collection containing news about a rumour that was immediately denied. In the second example, HEADY generalization improves over the most frequent pattern. In the third case, HEADY generates a good title from a noisy collection (containing different but related events). The examples also show that TopicSum is very effective in selecting a good human-generated headline for each collection. This opens the possibility of using TopicSum to automatically generate ROUGE references for future evaluations of abstractive methods.

6 Conclusions

We have presented HEADY, an abstractive headline generation system based on the generalization of syntactic patterns by means of a Noisy-OR Bayesian network. We evaluated the model both automatically and through human annotations. HEADY performs significantly better than a state-of-the-art open domain abstractive model (Filippova, 2010) in all evaluations, and is in par with human-generated headlines in terms of ROUGE scores. We have shown that it is possible to achieve high quality generation of news headlines in an open-domain, unsupervised setting by successfully exploiting syntactic and ontological information. The system relies on a standard NLP pipeline, requires no manual data annotation and can effectively scale to web-sized corpora.

For feature work, we plan to improve all components of HEADY in order to fill in the gap with the human-generated titles in terms of readability and informativeness. One of the directions in which we plan to move is the removal of the syntactic heuristics that currently enforce pattern well-formedness and to automatically learn the necessary transformations from the data.

Two other lines of work that we plan to explore are the possibility of personalizing the headlines to user interests (as stored in user profiles or expressed as user queries), and to investigate further applications of the Bayesian network of event patterns, such as its use for relation extraction and knowledge base population.

Acknowledgments

The research leading to these results has received funding from: the EU's 7th Framework Programme (FP7/2007-2013) under grant agreement number 257790; the Spanish Ministry of Science and Innovation's project Holopedia (TIN2010-21128-C02); and the Regional Government of Madrid's MA2VICMR (S2009/TIC1542). We would like to thank Katja Filippova and the anonymous reviewers for their insightful comments.

References

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, ACL '00, pages 318–325. Association for Computational Linguistics.

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 481–490. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, pages 3–3.

Domenic V Cicchetti. 1994. Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological Assessment*, 6(4):284.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31(1):399–429.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

William Coster and David Kauchak. 2011. Learning to simplify sentences using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9. Association for Computational Linguistics.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubi. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 1–8. Association for Computational Linguistics.

Micha Elsner and Deepak Santhanam. 2011. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 54–63. Association for Computational Linguistics.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 177–185. Association for Computational Linguistics.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics.

Nir Friedman and Moises Goldszmidt. 1996. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 252–262, San Francisco, CA. Morgan Kaufmann.

Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 180–187.

Pierre-Etienne Genest and Guy Lapalme. 2012. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, short papers*. Association for Computational Linguistics.

Dan Gillick. 2009. Sentence boundary detection and the problem with the us. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1152–1161. Association for Computational Linguistics.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

Tommi S. Jaakkola and Michael I. Jordan. 1999. Variational probabilistic inference and the QMR-DT Network. *Journal of Artificial Intelligence Research*, 10:291–322.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

Blackford Middleton, Michael Shwe, David Heckerman, Max Henrion, Eric Horvitz, Harold Lehmann, and Gregory Cooper. 1991. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. I. The probabilistic model and inference algorithms. *Methods of information in medicine*, 30(4):241–255, October.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518. ACM.

Tom Minka. 2002. judging significance from error bars. *CM U Tech R eport*.

Thahir P Mohamed, Estevam R Hruschka Jr, and Tom M Mitchell. 2011. Discovering relations between noun categories. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1447–1455. Association for Computational Linguistics.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: A taxonomy of relational patterns with semantic types. *EMNLP12*.

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011. Paraphrastic sentence compression with a character-based metric: Tightening without deletion. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 84–90. Association for Computational Linguistics.

Joakim Nivre. 2006. *Inductive Dependency Parsing*, volume 34 of *Text, Speech and Language Technology*. Springer.

Agnieszka Onisko, Marek J. Druzdzel, and Hanna Wasyluk. 2001. Learning Bayesian network parameters from small data sets: application of Noisy-OR gates. *International Journal of Approximated Reasoning*, 27(2):165–182.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of the NAACL-HLT 2012 Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9. Association for Computational Linguistics.

Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Tomáš Šingliar and Miloš Hauskrecht. 2006. Noisy-or component analysis and its application to link analysis. *J. Mach. Learn. Res.*, 7:2189–2213, December.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 409–420. Association for Computational Linguistics.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of The 23rd International Conference on Computational Linguistics*, pages 1353–1361.