Overcoming the Lack of Parallel Data in Sentence Compression

Katja Filippova and **Yasemin Altun**
Google
Brandschenkestr. 110
Zürich, 8004 Switzerland
katjaf|altun@google.com

Abstract

A major challenge in supervised sentence compression is making use of rich feature representations because of very scarce parallel data. We address this problem and present a method to automatically build a compression corpus with hundreds of thousands of instances on which deletion-based algorithms can be trained. In our corpus, the syntactic trees of the compressions are subtrees of their uncompressed counterparts, and hence supervised systems which require a structural alignment between the input and output can be successfully trained. We also extend an existing unsupervised compression method with a learning module. The new system uses structured prediction to learn from lexical, syntactic and other features. An evaluation with human raters shows that the presented data harvesting method indeed produces a parallel corpus of high quality. Also, the supervised system trained on this corpus gets high scores both from human raters and in an automatic evaluation setting, significantly outperforming a strong baseline.

1 Introduction and related work

Sentence compression is a paraphrasing task where the goal is to generate sentences shorter than given while preserving the essential content. A robust compression system would be useful for mobile devices as well as a module in an extractive summarization system (Mani, 2001). Although a compression may differ lexically and structurally from the source sentence, to date most systems are extractive and proceed by deleting words from the input (Knight & Marcu, 2000; Dorr et al., 2003; Turner & Charniak, 2005; Clarke & Lapata, 2008; Berg-Kirkpatrick et al., 2011, inter alia). To decide which words, dependencies or phrases can be dropped, (i) rule-based approaches (Grefenstette, 1998; Jing & McKeown, 2000; Dorr et al., 2003; Zajic et al., 2007), (ii) supervised models trained on parallel data (Knight & Marcu, 2000; Turner & Charniak, 2005; McDonald, 2006; Gillick & Favre, 2009; Galanis & Androutsopoulos, 2010, inter alia) and (iii) unsupervised methods which make use of statistics collected from non-parallel data (Hori & Furui, 2004; Zajic et al., 2007; Clarke & Lapata, 2008; Filippova & Strube, 2008) have been investigated. Since it is infeasible to manually devise a set of accurate deletion rules with high coverage, recent research has been devoted to developing statistical methods and possibly augmenting them with a few linguistic rules to improve output readability (Clarke & Lapata, 2008; Nomoto, 2009).

Supervised models. A major problem for supervised deletion-based systems is very limited amount of parallel data. Many approaches make use of a small portion of the Ziff-Davis corpus which has about 1K sentence-compression pairs[1]. Other main sources of training data are the two manually crafted compression corpora from the University of Edinburgh ("written" and "spoken", each approx. 1.4K pairs). Galanis & Androutsopoulos (2011) attempt at getting more parallel data by applying a deletion-based compressor together with an automatic para-

[1]The method of Galley & McKeown (2007) could benefit from a larger number of sentences.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1481–1491,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

phraser and generating multiple alternative compressions. To our knowledge, this extended data set has not yet been used for successful training of compression systems.

Scarce parallel data makes it hard to go beyond a small set of features and explore lexicalization. For example, Knight & Marcu (2000) only induce non-lexicalized CFG rules, many of which occurred only once in the training data. The features of McDonald (2006) are formulated exclusively in terms of syntactic categories. Berg-Kirkpatrick et al. (2011) have as few as 13 features to decide whether a constituent can be dropped. Galanis & Androutsopoulos (2010) use many features when deciding which branches of the input dependency tree can be pruned but require a reranker to select most fluent compressions from a pool of candidates generated in the pruning phase, many of which are ungrammatical.

Even further data limitations exist for the algorithms which operate on syntactic trees and reformulate the compression task as a tree pruning one (Nomoto, 2008; Filippova & Strube, 2008; Cohn & Lapata, 2009; Galanis & Androutsopoulos, 2010, inter alia). These methods are sensitive to alignment errors, their performance degrades if the syntactic structure of the compression is very different from that of the input. For example, see Nomoto's 2009 analysis of the poor performance of the T3 system of Cohn & Lapata (2009) when retrained on a corpus of loosely similar RSS feeds and news.

Unsupervised models. Few approaches require no training data at all. The model of Hori & Furui (2004) combines scores estimated from monolingual corpora to generate compressions of transcribed speech. Adopting an integer linear programming (ILP) framework, Clarke & Lapata (2008) use hand-crafted syntactic constraints and an ngram language model, trained on uncompressed sentences, to find best compressions. The model of Filippova & Strube (2008) also uses ILP but the problem is formulated over dependencies and not ngrams. Conditional probabilities and word counts collected from a large treebank are combined in an ad hoc manner to assess grammatical importance and informativeness of dependencies. Similarly, Woodsend & Lapata (2010) formulate an ILP problem to generate news story highlights using precomputed scores. Again, an ad hoc combination of the scores learned independently of the task is used in the objective function.

Contributions of this paper. Our work is motivated by the obvious need for a large parallel corpus of sentences and compressions on which extractive systems can be trained. Furthermore, we want the compressions in the corpus to be structurally very close to the input. Ideally, in every pair, the compression should correspond to a subtree of the input. To this end, our contributions are three-fold:

- We describe an automatic procedure of constructing a parallel corpus of 250,000 sentence-compression pairs such that the dependency tree of the compression is a subtree of the source tree. An evaluation with human raters demonstrates high quality of the parallel data in terms of readability and informativeness.

- We successfully apply the acquired data to train a novel supervised compression system which produces readable and informative compressions without employing a separate reranker. In particular, we start with the unsupervised method of Filippova & Strube (2008) and replace the ad hoc edge weighting with a linear function over a rich feature representation. The parameter vector is learned from our corpus specifically for the compression task using structured prediction (Collins, 2002). The new system significantly outperforms the baseline and hence provides further evidence for the utility of the parallel data.

- We demonstrate that sparse lexical features are very useful for sentence compression, and that a large parallel corpus is a requirement for applying them successfully.

The compression framework we adopt and the unsupervised baseline are introduced in Section 2, the training algorithm for learning edge weights from parallel data is described in Section 3. In Section 4 we explain how to obtain the data and present an evaluation of its quality. In Section 5 we compare the baseline with our system and report the results of an experiment with humans as well as the results of an automatic evaluation.

2 Framework and baseline

We adopt the unsupervised compression framework of Filippova & Strube (2008) as our baseline and extend it to a supervised structured prediction problem. In the experiments reported by Filippova & Strube (2008), the system was evaluated on the Edinburgh corpora. It achieved an F-score (Riezler et al., 2003) higher than reported by other systems on the same data under an aggressive compression rate and thus presents a competitive baseline.

Tree pruning as optimization. In this framework, compressions are obtained by deleting edges of the source dependency structure so that (1) the retained edges form a valid syntactic tree, and (2) their total edge weight is maximized. The objective function is defined over set $X = \{x_e, e \in E\}$ of binary variables, corresponding to the set E of the source edges, subject to the structural and length constraints,

$$f(X) = \sum_{e \in E} x_e \times w(e) \tag{1}$$

Here, $w(e)$ denotes the weight of edge e. This constrained optimization problem is solved under the tree structure and length constraints using ILP. If x_e is resolved to 1, the respective edge is retained, otherwise it is deleted. The tree structure constraints enforce at most one parent for every node and structure connectivity (i.e., no disconnected subtrees). Given that $length(node(e))$ denotes the length of the node to which edge e points and α is the maximum permitted length for the compression, the length constraint is simply

$$\sum_{e \in E} x_e \times length(node(e)) \leq \alpha \tag{2}$$

Word limit is used in the original paper, whereas we use character length which is more appropriate for system comparisons (Napoles et al., 2011). If uniform weights are used in Eq. (1), the optimal solution would correspond to a subtree covering as many edges as possible while keeping the compression length under given limit.

The solution to the surface realization problem (Belz et al., 2011) is standard: the words in the compression subtree are put in the same order they are found in the source.

Due to space limitations, we refer the reader to (Filippova & Strube, 2008) for a detailed description on the method. Essential for the present discussion is that source dependency trees are transformed to dependency graphs in that (1) auxiliary, determiner, preposition, negation and possessive nodes are collapsed with their heads; (2) prepositions replace labels on the edges to their arguments; (3) the dummy root node is connected with every inflected verb. Figures 1(a)-1(b) illustrate most of the transformations. The transformations are deterministic and reversible, they can be implemented in a single top-down tree traversal[2].

The set E of edges in Eq. (1) is thus the set of edges of the *transformed dependency graph*, like in Fig. 1(b). A benefit of the transformations is that function words and negation appear in the compression if and only if their head words are present. Hence no separate constraints are required to ensure that negation or a determiner is preserved. The dummy root node makes constraint formulation easier and also allows for the generation of compressions from any finite clause of the source.

The described pruning optimization framework is used both for the unsupervised baseline and for our supervised system. The difference between the baseline and our system is in how edge weights, $w(e)$'s in Eq. (1), are instantiated.

Baseline edge weights. The precomputed edge weights reflect syntactic importance as well as informativeness of the nodes they point to. Given edge e from head node h to node n, the edge weight is the product of the syntactic and the informativeness weights,

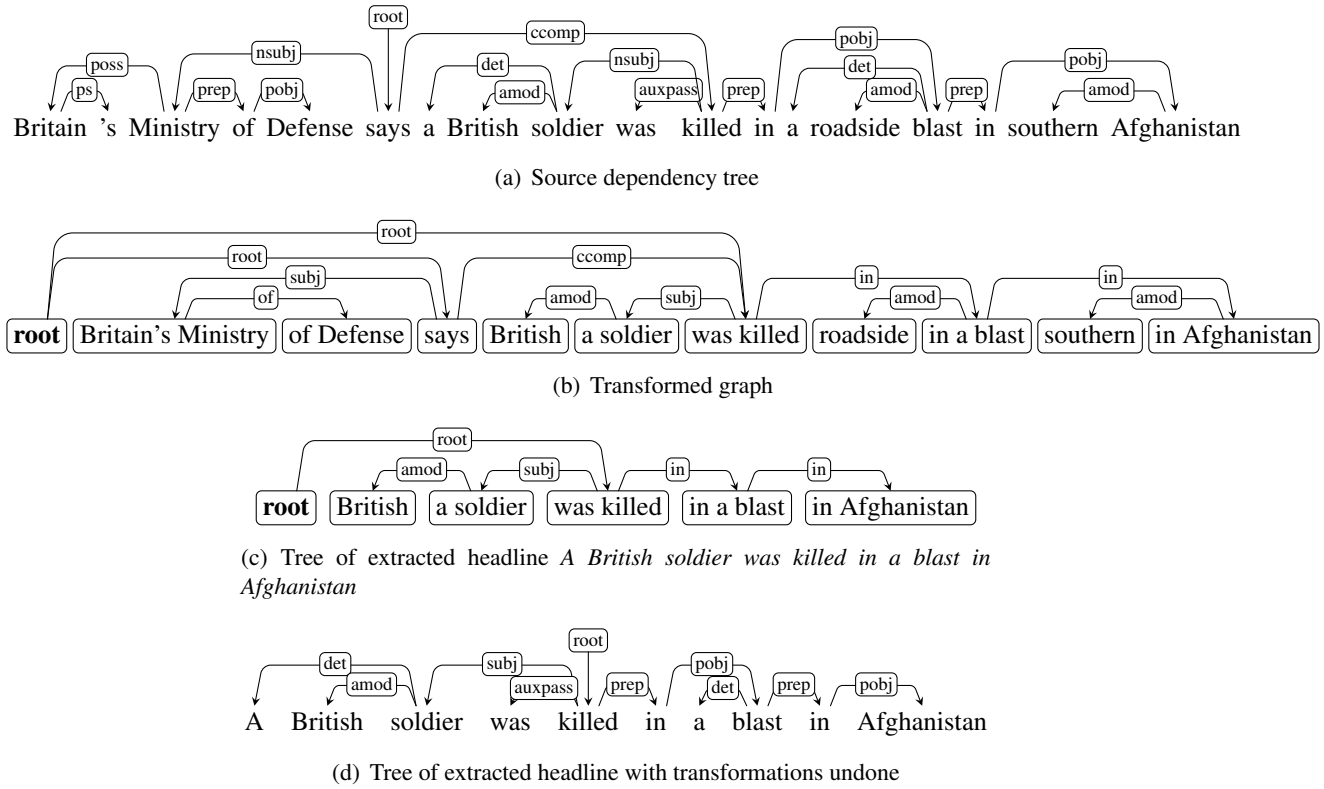
$$w(e) = w_{\text{synt}}(e) \times w_{\text{info}}(e) \tag{3}$$

The syntactic weight is defined as

$$w_{\text{synt}}(e) = P(label(e)|lemma(h)) \tag{4}$$

For example, verb *kill* may have multiple arguments realized with dependency labels *subj*, *dobj*, *in*, etc. However, these argument labels are not equally likely, e.g., $P(\textit{subj}|\textit{kill}) > P(\textit{in}|\textit{kill})$. When forced to prune an edge, the system would prefer to keep

[2]Some of the transformations are comparable to what is implemented in the Stanford parser (de Marneffe et al., 2006).



(a) Source dependency tree

(b) Transformed graph

(c) Tree of extracted headline *A British soldier was killed in a blast in Afghanistan*

(d) Tree of extracted headline with transformations undone

Figure 1: Source, transformed and extracted trees given headline *British soldier killed in Afghanistan*

the subject edge over the preposition-in edge since it contributes more weight to the objective function.

The informativeness score is inspired by Woodsend & Lapata (2012) and is defined as

$$w_{\text{info}}(e) = \frac{P_{\text{headline}}(lemma(n))}{P_{\text{article}}(lemma(n))} \quad (5)$$

This weight tells us how likely it is that a word from an article appears in the headline. For example, given two edges one of which points to verb *say* and another one to verb *kill*, the latter would be preferred over the former because *kill* is more "headliny" than *say*. When collecting counts for the syntactic and informativeness scores, we used 9M news articles crawled from the Internet, much more than Filippova & Strube (2008). As a result our estimates are probably more accurate than theirs.

Although both w_{synt} and w_{info} have a meaningful interpretation, there is no guarantee that product is the best way to combine the two when assigning edge weights. Also, it is unclear how to integrate other signals, such as distance to the root, node length or information about the siblings, which presumably all play a role in determining the overall edge importance.

3 Learning edge weights

Our supervised system differs from the unsupervised baseline in that instead of relying on precomputed scores, we define edge weight $w(e)$ in Eq. (1) with a linear function over a feature representation,

$$w(e) = \mathbf{w} \cdot \mathbf{f}(e) \quad (6)$$

Here $\mathbf{f}(e)$ is a vector of binary variables for every feature from the set of all possible but very infrequent features in the training set. $\mathbf{f}(e)$ has 1 for every feature extracted for edge *e* and zero otherwise.

Table 1 gives an overview of the feature types we use (edge *e* points from head *h* to node *n*). Note that syntactic, structural and semantic features are closed-class. For all the structural features but *char_length*, seven is used as maximum possible value; all possible character lengths are bucketed into six classes. All the features are local – for a given edge, contextual information is included about

syntactic	*label(e)*; for *e** to *h*, *label(e*)*; *pos(h)*; *pos(n)*
structural	*depth(n)*; *#children(n)*; *#children(h)*; *char_length(n)*; *#words_in(n)*
semantic	*NE_tag(h)*; *NE_tag(n)*; *is_negated(n)*
lexical	*lemma(n)*; *lemma(h)-label(e)*; for *e** to *n*'s siblings, *lemma(h)-label(e*)*

Table 1: Types of features extracted for edge *e* from *h* to *n*

the head and the target nodes, and the siblings as well as the children of the latter. The negation feature is only applicable to verb nodes which contain a negative particle, like *not*, after the tree transformations. Lexical features which combine lemmas and syntactic labels are inspired by the unsupervised baseline and are very sparse.

In what follows, our assumption is that we have a compression corpus at our disposal where for every input sentence there is a correct "oracle" compression such that its transformed parse tree matches a subtree of the transformed input graph. Given such a corpus, we can apply structured prediction methods to learn the parameter vector $\mathbf{w}$. In our study we employ an averaged variant of online structured perceptron (Collins, 2002). In the context of sentence fusion, a similar dependency structure pruning framework and a similar learning approach was adopted by Elsner & Santhanam (2011).

At every iteration, for every input graph, we find the optimal solution with ILP under the current parameter vector $\mathbf{w}$. The maximum permitted compression length is set to be the same as the length of the oracle compression. Since the oracle compression is a subtree of the input graph, it represents a feasible solution for ILP. The parameter vector is updated if there is a mismatch between the predicted and the oracle sets of edges for all the features with a non-zero net count. More formally, given an input graph with the set of edges E, oracle compression $C \subset E$ and compression $C_t \subseteq E$ predicted at iteration t , the parameter update vector at $t+1$ is given by

$$\mathbf{w_{t+1}} = \mathbf{w_t} + \sum_{e \in C \setminus C_t} \mathbf{f}(e) - \sum_{e \in C_t \setminus C} \mathbf{f}(e) \quad (7)$$

$\mathbf{w}$ is averaged over all the $\mathbf{w_t}$'s so that features whose weight fluctuated a lot during training are penalized (Freund & Shapire, 1999).

Of course, training a model with a large number of features, such as a lexicalized model, is only possible if there is a large compression corpus where the dependency tree of the compression is a subtree of the source sentence. In the next section we introduce our method of getting a sufficient amount of such data.

4 Acquiring parallel data automatically

In this section we explain how we obtained a parallel corpus of sentences and compressions. The underlying idea is to harvest news articles from the Internet where the headline appears to be similar to the first sentence and use it to find an extractive compression of the sentence.

Collecting headline-sentence pairs. Using a news crawler, we collected a corpus of news articles in English from the Internet. Similarly to previous work (Dolan et al., 2004; Wubben et al., 2009; Bejan & Harabagiu, 2010, inter alia), the Google News service[3] was used to identify news. From every article, the headline and the first sentence, which are known to be semantically similar (Dorr et al., 2003), were extracted. Predictably, very few headlines are extractive compressions of the first sentence, therefore simply looking for pairs where the headline is a subsequence of the words from the first sentence would not solve the problem of getting a large amount of parallel data. Importantly, headlines are syntactically quite different from "normal" sentences. For example, they may have no main verb, omit determiners and appear incomplete, making it hard for a supervised deletion-based system to learn useful rules. Moreover, we observed poor parsing accuracy for headlines which would make syntactic annotations for headlines hardly useful.

Thus, instead of taking the headline as it is, we use it to find a proper extractive compression of the sen-

[3] `http://news.google.com`, Jan-Dec 2012.

tence by matching lemmas of content words (*nouns, verbs, adjectives, adverbs*) and coreference IDs of entities from the headline with those of the sentence. The exact procedure is as follows (H, S and T stand for *headline*, *sentence* and *transformed graph* of the sentence):

PREPROCESSING H and S are preprocessed in a standard way: tokenized, lemmatized, PoS and NE tagged. Additionally, S is parsed with a dependency parser (Nivre, 2006) and transformed as described in Section 2 to obtain T. Finally, pronominal anaphora is resolved in S. Recall that S is the first sentence, so the antecedent must be located in a preceding, higher-level clause.

FILTERING To restrict the corpus to grammatical and informative headlines, we implemented a cascade of filters. Pair (H, S) is discarded if any of the questions in Table 2 is answered positively.

Is H a question?
Is H or S too short? (less than four word tokens)
Is H about as long as S? (min ratio: 1.5)
Does H lack a verb?
Does H begin with a verb?
Is there a *noun, verb, adj, adv* lemma from H not found in S?
Are the *noun, verb, adj, adv* lemmas from H found in S in a different order?

Table 2: Filters applied to candidate pair (H, S)

MATCHING Given the content words of H, a subset of nodes in T is selected based on lemma or coreference identity of the main (head) word in the nodes. For example, the main word of a collapsed node in T, which covers two words *was killed*, is *killed*; *was* is its child attached with label *aux* in the untransformed parse tree. This node is marked if H contains word *killed* or *killing* because of the lemma identity. In some cases there are multiple possible matches. For example, given S *Barack Obama said he will attend G20* and H mentioning *Obama*, both *Barack Obama* and *he* nodes are marked in T. Once all the nodes in T which match content words and entities from H are identified, a minimum subtree covering these nodes is found such that every word or entity from H occurs as many times in T as in H. So if *H* mentions *Obama* only once, then either *Barack Obama* or *he* must be covered by the subtree but not both. This minimum subtree corresponds to an **extractive headline**, H⋆, which we generate by ordering the surface forms of all the words in the subtree nodes by their offsets in S. Finally, the character length of H⋆ is compared with the length of H. If H⋆ is much longer than H, the pair (H, S) is discarded (max ratio 1.5).

As an illustration to the procedure, consider the example from Figure 1 with the extracted headline and its tree presented in Figure 1(c). Given the headline *British soldier killed in Afghanistan*, the extracted headline would be *A British soldier was killed in a blast in Afghanistan*. The lemmas *british, soldier, kill, afghanistan* from the headline match the nodes *British, a soldier, was killed, in Afghanistan* in the transformed graph. The node *in a blast* is added because it is on the path from *was killed* to *in Afghanistan*. Of course, it is possible to deterministically undo the transformations in order to obtain a standard dependency tree. In this case the extracted headline would still correspond to a subtree of the input (compare Fig. 1(d) with Fig. 1(a)). Also note that a similar procedure can be implemented for constituency parses.

The resulting corpus consists of 250K tuples (S, T, H, H⋆), Appendix provides more examples of source sentences, original headlines and extracted headlines. We did not attempt to tune the values for minimum/maximum length and ratio – lower thresholds may have produced comparable results.

Evaluating data quality. The described procedure produces a comparatively large compression corpus but how good are automatically constructed compressions? To answer this question, we randomly selected 50 tuples from the corpus and set up an experiment with human raters to validate and assess data quality in terms of *readability*[4] and *informativeness*[5] which are standard measures of compression quality (Clarke & Lapata, 2006). Raters were asked to read a sentence and a compression (original H or extracted H⋆ headline) and then rate the compression on two five-point scales. Three ratings were collected for every item. Table 3 gives

[4] Also called *grammaticality* and *fluency*.

[5] Also called *importance* and *representativeness*.

average ratings with standard deviation.

	AVG read	*AVG info*
ORIG. HEADLINE	4.36 (0.75)	3.86 (0.79)
EXTR. HEADLINE	4.26 (1.01)	3.70 (1.04)

Table 3: Results for two kinds of headlines

In terms of readability and informativeness the extracted headlines are comparable with human-written ones: at 95% confidence there is no statistically significant difference between the two.

Encouraged by the results of the validation experiment we proceeded to our next question: Can a supervised compression system be successfully trained on this corpus?

5 System evaluation and discussion

From the corpus of 250K tuples we used 100K to get pairs of *extracted* headlines and sentences for training (on the development set we did not observe much improvement from using more training data), 250 for development and the rest for testing. We ran the learning algorithm for 20 iterations, checking the performance on the development set. Features which applied to less than 20 edges were pruned, the size of the feature set is about 28K.

5.1 Evaluation with humans

50 pairs of *original* headlines and sentences (different from the data validation set in Sec. 4) were randomly selected for an evaluation with humans from the test data. As in the data quality validation experiment, we asked raters to assess the readability and informativeness of proposed compressions for the unsupervised system, our system and human-written headlines. The latter provide us with upper bounds on the evaluation criteria. Three ratings per item per parameter were collected. To get comparable results, the unsupervised and our systems used the same compression rate: for both, the requested maximum length was set to the length of the headline. Table 4 summarizes the results.

The results indicate that the trained model significantly outperforms the unsupervised system, getting particularly good marks for readability. The difference in readability between our system and *original* headlines is not statistically significant. Note that

	AVG read	*AVG info*
ORIG. HEADLINE	$4.66^{\dagger}$	$4.10^{\dagger\ddagger}$
OUR SYSTEM	$4.30^{\dagger}$	$3.52^{\dagger}$
UNSUP. SYSTEM	3.70	2.70

Table 4: Results for the systems and original headline: † and ‡ stand for *significantly better than Unsupervised* and *Our system at 95% confidence*, respectively

the unsupervised baseline is also capable of generating readable compressions but does a much poorer job in selecting most important information. Our trained model successfully learned to optimize both scores. We refer the reader to Appendix for input and compression examples. Note that the ratings for the human-written headlines in this experiment are slightly different from the ratings in the data validation experiment because a different data sample was used.

5.2 Automatic evaluation

Our automatic evaluation had the goal of explicitly addressing two relevant questions related to our claims about (1) the benefits of having a large parallel corpus and (2) employing a supervised approach with a rich feature representation.

1. Our primary motivation for collecting parallel data has been that having access to sparse lexical features, which considerably increase the feature space, would benefit compression systems. But is it really the case for sentence compression? Can a comparable performance be achieved with a closed, moderately sized set of dense, non-lexical features? If yes, then a large compression corpus is probably not needed. Furthermore, to demonstrate that a large corpus is not only sufficient but also necessary to learn weights for thousands of features, we need to compare the performance of the system when trained on the full data set and a small portion of it.

2. The syntactic and informativeness scores in Eq. (3) were calculated over millions of news articles and do provide us with meaninful statistics (see Sec. 2). Is there any benefit in replacing those scores with weights learned for

their feature counterparts? Recall that one of our feature types in Table 1 is the concatenation of *lemma(h)* (parent lemma) and *label(e)* which relies on the same information as $w_{\text{synt}} = P(label(e)|lemma(h))$. The feature counterpart of w_{info} defined in Eq. (5) is *lemma(n)*–the lemma of the node to which edge points. How would the supervised system perform against the unsupervised one, if it only extracted features of these two types?

To answer these questions, we sampled 1,000 tuples from the unused test data and measured F1 score (Riezler et al., 2003) by comparing the trees of the generated compression and the "correct", extracted headline. The systems we compared are the unsupervised baseline (UNSUP. SYSTEM) and the supervised model trained on three kinds of feature sets: (1) SYNT-INFO FEATURES, corresponding to the supervised training of the unsupervised baseline model (i.e., *lemma(h)-label(e)* and *lemma(n)*); (2) NON-LEX FEATURES, corresponding to a dense, non-lexical feature representation (i.e., all the feature types from Table 1 excluding the three involving *lemma*s); (3) ALL FEATURES (same as OUR SYSTEM). Additionally, we trained the system on 10% of the data–10K as opposed to 100K tuples, ALL FEATURES (10K)–for 20 iterations ignoring features which applied to less than three edges[6]. As before, the same compression rate was used for all the systems. The results are summarized in Table 5.

	F1 score	*#features*
UNSUP. SYSTEM	52.3	N.A.
SYNT-INFO FEATURES	75.0	12,490
NON-LEX FEATURES	79.6	330
ALL FEATURES	84.3	27,813
ALL FEATURES (10K)	81.4	22,529

Table 5: Results for the unsupervised baseline and the supervised system trained on three kinds of feature sets

Clearly, having more features, lexicalized and unlexicalized, is important: there is a significant improvement in going beyond the closed set of 330 non-lexical features to all, from 79.6 to 84.3 points. Moreover, successful training requires a large corpus since the performance of the system degrades if only 10K training instances are used. Note that this number already exceeds all the existing compression corpora taken together. Hence, sparse lexical features are useful for compression and a large parallel corpus is a requirement for successful supervised training.

Concerning our second question, learning feature weights from the data produces significantly better results than the hand-crafted way of making use of the same information, even if a much larger data set is used to collect statistics. We observed a dramatic increase from 52.3 to 75.0 points. Thus, we may conclude that training with dense and sparse features directly from data definitely improves the performance of the dependency pruning system.

5.3 Discussion

It is important to note that the data we used is challenging: first sentences in news articles tend to be long, in fact longer than other news sentences, which implies less reliable syntactic analysis and noisier input to the syntax-based systems. In the test set we used for the evaluation with humans, the mean sentence length is 165 characters. The average compression rate in characters is 0.46 ± 0.16 which is quite aggressive[7]. Recall that we used the very same framework for the unsupervised baseline and our system as well as the same compression rate. All the preprocessing errors affect both systems equally and the comparison of the two is fair. Predictably, wrong syntactic parses significantly increase chances of an ungrammatical compression, and parser errors seem to be a major source of readability deficiencies.

A property of the described compression framework is that a desired compression length is expected to be provided by the user. This can be seen both as a strength and as a weakness, depending on the application. In a scenario where mobile devices with a limited screen size are used, or in a summarization scenario where a total summary length is provided (see the DUC/TAC guidelines[8]), being able

[6]Recall from the beginning of the section that for the full (100K) training set the threshold was set to 20 with no tuning. For the 10K training set, we tried values of two, three, five and varied the number of iterations. The result we report is the highest we could get for 10K.

[7]We follow the standard terminology where smaller values imply *shorter* compressions.

[8]http://www.nist.gov/tac/

to specify a length is definitely an advantage. However, one can also think of other applications where the user does not have a strict length constraint but wants the text to be somewhat shorter. In this case, a reranker which compares compressions generated for a range of possible lengths can be employed to find a single compression (e.g., mean edge weight in the solution or a language model-based score).

6 Conclusions

We have addressed a major problem for supervised extractive compression models – the lack of a large parallel corpus. To this end, we presented a method to automatically build such a corpus from web documents available on the Internet. An evaluation with humans demonstrates that the quality of the corpus is high – the compressions are grammatical and informative. We also significantly improved a competitive unsupervised method achieving high readability and informativeness scores by incorporating thousands of features and learning the feature weights from our corpus. This result further confirms the practical utility of the automatically obtained data. We have shown that employing lexical features is important for sentence compression, and that our supervised module can successfully learn their weights from the corpus. To our knowledge, we are the first to empirically demonstrate that sparse features are useful for compression and that a large parallel corpus is a requirement for a successful learning of their weights. We believe that other supervised deletion-based systems can benefit from our work.

Acknowledgements: The authors are thankful to the EMNLP reviewers for their feedback and suggestions.

Appendix

The appendix presents examples of source sentences (S), original headlines (H), extracted headlines (H*), unsupervised baseline (U) and our system (O) compressions.

References

Bejan, C. & S. Harabagiu (2010). Unsupervised event coreference resolution with rich linguistic features. In *Proc. of ACL-10*, pp. 1412–1422.

Belz, A., M. White, D. Espinosa, E. Kow, D. Hogan & A. Stent (2011). The first surface realization shared task: Overview and evaluation results. In *Proc. of ENLG-11*, pp. 217–226.

Berg-Kirkpatrick, T., D. Gillick & D. Klein (2011). Jointly learning to extract and compress. In *Proc. of ACL-11*.

Clarke, J. & M. Lapata (2006). Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proc. of COLING-ACL-06*, pp. 377–385.

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

Cohn, T. & M. Lapata (2009). Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

Collins, M. (2002). Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP-02*, pp. 1–8.

de Marneffe, M.-C., B. MacCartney & C. D. Manning (2006). Generating typed dependency parses from phrase structure parses. In *Proc. of LREC-06*, pp. 449–454.

Dolan, B., C. Quirk & C. Brokett (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics,* Geneva, Switzerland, 23–27 August 2004, pp. 350–356.

Dorr, B., D. Zajic & R. Schwartz (2003). Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the Text Summarization Workshop at HLT-NAACL-03,* Edmonton, Alberta, Canada, 2003, pp. 1–8.

S	Country star Sara Evans has married former University of Alabama quarterback Jay Barker.
H	Country star Sara Evans marries
H*	Country star Sara Evans has married
U	Sara Evans has married Jay Barker
O	Sara Evans has married Jay Barker
S	Intel would be building car batteries, expanding its business beyond its core strength, the company said in a statement
H	Intel to build car batteries
H*	Intel would be building car batteries
U	would be building the company said
O	Intel would be building car batteries
S	A New Orleans Saints team spokesman says tight end Jeremy Shockey was taken to a hospital but is doing fine.
H	Spokesman: Shockey taken to hospital, doing fine
H*	spokesman says Jeremy Shockey was taken to a hospital but is doing fine
U	A New Orleans Saints team spokesman says Jeremy Shockey was taken
O	tight end Jeremy Shockey was taken to a hospital but is doing fine
S	President Obama declared a major disaster exists in the State of Florida and ordered Federal aid to supplement State and local recovery efforts in the area struck by severe storms, flooding, tornadoes, and straight-line winds beginning on May 17, 2009, and continuing.
H	President Obama declares major disaster exists in the State of Florida
H*	President Obama declared a major disaster exists in the State of Florida
U	President Obama declared a major disaster exists and ordered Federal aid
O	President Obama declared a major disaster exists in the State of Florida
S	Regulators Friday shut down a small Florida bank, bringing to 119 the number of US bank failures this year amid mounting loan defaults.
H	Regulators shut down small Florida bank
H*	Regulators shut down a small Florida bank
U	shut down bringing the number of failures
O	Regulators shut down a small Florida bank
S	Three men were arrested Wednesday night and Dayton police said their arrests are in connection to a west Dayton bank robbery.
H	3 men arrested in connection with Bank robbery
H*	Three men were arrested are in connection to a bank robbery
U	were arrested and Dayton police said their arrests are
O	Three men were arrested and police said their arrests are
S	The government and the social partners will resume the talks on the introduction of the so-called crisis tax, which will be levied on all salaries, pensions and incomes over HRK 3,000.
H	Government, social partners to resume talks on introduction of “crisis” tax.
H*	The government and the social partners will resume the talks on the introduction of the crisis tax
U	The government will resume the talks on the introduction of the crisis tax which will be levied
O	The government and the social partners will resume the talks on the introduction of the crisis tax
S	England star David Beckham may have the chance to return to AC Milan after the Italian club’s coach said he was open to his move on Sunday.
H	Beckham has chance of returning to Milan
H*	David Beckham may have the chance to return to AC Milan
U	David Beckham may have the chance to return said star was
O	David Beckham may have the chance to return to AC Milan
S	Eastern Health and its insurance company have accepted liability for some patients involved in the breast cancer testing scandal, according to a statement released Friday afternoon.
H	Eastern Health accepts liability for some patients
H*	Eastern Health have accepted liability for some patients
U	Health have accepted liability according to a statement
O	Eastern Health have accepted liability for some patients
S	Frontier Communications Corp., a provider of phone, TV and Internet services, said Thursday it has started a cash tender offer to purchase up to $700 million of its notes.
H	Frontier Communications starts tender offer for up to $700 million of notes
H*	Frontier Communications has started a tender offer to purchase $700 million of its notes
U	Frontier Communications said Thursday a provider has started a tender offer
O	Frontier Communications has started a tender offer to purchase $700 million of its notes

Elsner, M. & D. Santhanam (2011). Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-to-text Generation,* Prtland, OR, June 24 2011, pp. 54–63.

Filippova, K. & M. Strube (2008). Dependency tree based sentence compression. In *Proc. of INLG-08*, pp. 25–32.

Freund, Y. & R. E. Shapire (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296.

Galanis, D. & I. Androutsopoulos (2010). An extractive supervised two-stage method for sentence compression. In *Proc. of NAACL-HLT-10*, pp. 885–893.

Galanis, D. & I. Androutsopoulos (2011). A new sentence compression dataset and its use in an abstractive generate-and-rank sentence compressor. In *Proc. of UCNLG+Eval-11*, pp. 1–11.

Galley, M. & K. R. McKeown (2007). Lexicalized Markov grammars for sentence compression. In *Proc. of NAACL-HLT-07*, pp. 180–187.

Gillick, D. & B. Favre (2009). A scalable global model for summarization. In *ILP for NLP-09*, pp. 10–18.

Grefenstette, G. (1998). Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Working Notes of the Workshop on Intelligent Text Summarization,* Palo Alto, Cal., 23 March 1998, pp. 111–117.

Hori, C. & S. Furui (2004). Speech summarization: An approach through word extraction and a method for evaluation. *IEEE Transactions on Information and Systems*, E87-D(1):15–25.

Jing, H. & K. McKeown (2000). Cut and paste based text summarization. In *Proc. of NAACL-00*, pp. 178–185.

Knight, K. & D. Marcu (2000). Statistics-based summarization – step one: Sentence compression. In *Proc. of AAAI-00*, pp. 703–711.

Mani, I. (2001). *Automatic Summarization.* Amsterdam, Philadelphia: John Benjamins.

McDonald, R. (2006). Discriminative sentence compression with soft syntactic evidence. In *Proc. of EACL-06*, pp. 297–304.

Napoles, C., C. Callison-Burch, J. Ganitkevitch & B. 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,* Prtland, OR, June 24 2011, pp. 84–90.

Nivre, J. (2006). *Inductive Dependency Parsing.* Springer.

Nomoto, T. (2008). A generic sentence trimmer with CRFs. In *Proc. of ACL-HLT-08*, pp. 299–307.

Nomoto, T. (2009). A comparison of model free versus model intensive approaches to sentence compression. In *Proc. of EMNLP-09*, pp. 391–399.

Riezler, S., T. H. King, R. Crouch & A. Zaenen (2003). Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. In *Proc. of HLT-NAACL-03*, pp. 118–125.

Turner, J. & E. Charniak (2005). Supervised and unsupervised learning for sentence compression. In *Proc. of ACL-05*, pp. 290–297.

Woodsend, K. & M. Lapata (2010). Automatic generation of story highlights. In *Proc. of ACL-10*, pp. 565–574.

Woodsend, K. & M. Lapata (2012). Multiple aspect summarization using Integer Linear Programming. In *Proc. of EMNLP-12*, pp. 233–243.

Wubben, S., A. van den Bosch, E. Krahmer & E. Marsi (2009). Clustering and matching headlines for automatic paraphrase acquisition. In *Proc. of ENLG-09*, pp. 122–125.

Zajic, D., B. J. Dorr, J. Lin & R. Schwartz (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management, Special Issue on Text Summarization*, 43(6):1549–1570.