Sentence Compression with Semantic Role Constraints

Katsumasa Yoshikawa
Precision and Intelligence Laboratory,
Tokyo Institute of Technology, Japan
IBM Research-Tokyo, IBM Japan, Ltd.
katsumasay@gmail.com

Ryu Iida
Department of Computer Science,
Tokyo Institute of Technology, Japan
ryu-i@cl.cs.titech.ac.jp

Tsutomu Hirao
NTT Communication Science Laboratories,
NTT Corporation, Japan
hirao.tsutomu@lab.ntt.co.jp

Manabu Okumura
Precision and Intelligence Laboratory,
Tokyo Institute of Technology, Japan
oku@lr.pi.titech.ac.jp

Abstract

For sentence compression, we propose new semantic constraints to directly capture the relations between a predicate and its arguments, whereas the existing approaches have focused on relatively shallow linguistic properties, such as lexical and syntactic information. These constraints are based on semantic roles and superior to the constraints of syntactic dependencies. Our empirical evaluation on the Written News Compression Corpus (Clarke and Lapata, 2008) demonstrates that our system achieves results comparable to other state-of-the-art techniques.

1 Introduction

Recent work in document summarization do not only extract sentences but also compress sentences. Sentence compression enables summarizers to reduce the redundancy in sentences and generate informative summaries beyond the extractive summarization systems (Knight and Marcu, 2002). Conventional approaches to sentence compression exploit various linguistic properties based on lexical information and syntactic dependencies (McDonald, 2006; Clarke and Lapata, 2008; Cohn and Lapata, 2008; Galanis and Androutsopoulos, 2010).

In contrast, our approach utilizes another property based on semantic roles (SRs) which improves weaknesses of syntactic dependencies. Syntactic dependencies are not sufficient to compress some complex sentences with coordination, with passive voice, and with an auxiliary verb. Figure 1 shows an example with a coordination structure. 1

1This example is from Written News Compression Corpus (http://jamesclarke.net/research/resources).

In this example, a SR labeler annotated that Harari is an A0 argument of left and an A1 argument of became. Harari is syntactically dependent on left—SBJ(left-2, Harari-1). However, Harari is not dependent on became and we are hence unable to utilize a dependency relation between Harari and became directly. SRs allow us to model the relations between a predicate and its arguments in a direct fashion.

SR constraints are also advantageous in that we can compress sentences with semantic information. In Figure 1, became has three arguments, Harari as A1, businessman as A2, and shortly afterward as AM-TMP. As shown in this example, shortly afterward can be omitted (shaded boxes). In general, modifier arguments like AM-TMP or AM-LOC are more likely to be reduced than complement cases like A0-A4. We can implement such properties by SR constraints.

Liu and Gildea (2010) suggests that SR features contribute to generating more readable sentence in machine translation. We expect that SR features also help our system to improve readability in sentence compression and summarization.

2 Why are Semantic Roles Useful for Compressing Sentences?

Before describing our system, we show the statistics in terms of predicates, arguments and their rela-
on building a set of formulae called Markov Logic Network (MLN). So, in this section, we describe our proposed MLN in detail.

### 3.1 Proposed Markov Logic Network

First, let us define our MLN predicates. We summarize the MLN predicates in Table 2. We have only one hidden MLN predicate, \( \text{inComp}(i) \) which models the decision we need to make: whether a token \( i \) is in compression or not. The other MLN predicates are called \( \text{observed} \) which provide features. With our MLN predicates defined, we can now go on to incorporate our intuition about the task using weighted first-order logic formulae. We define SR constraints and the other formulae in Sections 3.1.1 and 3.1.2, respectively.

#### 3.1.1 Semantic Role Constraints

Semantic role labeling generally includes the three subtasks: predicate identification; argument role labeling; sense disambiguation. Our model exploits the results of predicate identification and argument role labeling. 4 \( \text{pred}(i) \) and \( \text{role}(i,j,r) \) indicate the results of predicate identification and role labeling, respectively.

First, the formula describing a local property of a predicate is

\[
\text{pred}(i) \Rightarrow \text{inComp}(i)
\]

which denotes that, if token \( i \) is a predicate then \( i \) is in compression. A formula with exact one hidden predicate is called local formula.

A predicate is not always in compression. The formula reducing some predicates is

\[
\text{pred}(i) \land \text{height}(i, +n) \Rightarrow \neg\text{inComp}(i)
\]

which implies that a predicate \( i \) is not in compression with \( n \) height in a dependency tree. Note the + notation indicates that the MLN contains one instance of the rule, with a separate weight, for each assignment of the variables with a plus sign.

As mentioned earlier, our SR constraints model the difference of the survival rate of role labels in compression. Such SR constraints are encoded as:

\[
\text{role}(i,j, +r) \land \text{inComp}(i) \Rightarrow \text{inComp}(j)
\]

\[
\text{role}(i,j, +r) \land \neg\text{inComp}(i) \Rightarrow \neg\text{inComp}(j)
\]

which represent that, if a predicate \( i \) is (not) in compression, then its argument \( j \) is (not) also in with

| Label     | In Compression / Total | Ratio |
|-----------|------------------------|-------|
| A0        | 1454 / 1607            | 0.905 |
| A1        | 1916 / 2208            | 0.868 |
| A2        | 427 / 490              | 0.871 |
| AM-TMP    | 261 / 488              | 0.535 |
| AM-LOC    | 134 / 214              | 0.626 |
| AM-ADV    | 115 / 213              | 0.544 |
| AM-DIS    | 8 / 8                  | 0.094 |

Table 1: Statistics of Arguments in Compression in the Written News Compression (WNC) Corpus. It has 82 documents (1,629 sentences). We divided them into three: 55 documents are used for training (1106 sentences); 10 for development (184 sentences); 17 for testing (339 sentences).

Our investigation was held in training data. There are 3137 verbal predicates and 7852 unique arguments. We performed SR labeling by LTH (Johansson and Nugues, 2008), an SR labeler for CoNLL-2008 shared task. Based on the SR labels annotated by LTH, we investigated that, for all predicates in compression, how many their arguments were also in. Table 1 shows the survival ratio of main arguments in compression. Labels A0, A1, and A2 are complement case roles and over 85% of them survive with their predicates. On the other hand, for modifier arguments (AM-X), survival ratios are down to lower than 65%. Our SR constraints implement the difference of survival ratios by SR labels. Note that dependency labels SBJ and OBJ generally correspond to SR labels A0 and A1, respectively. But their total numbers are 777 / 919 (SBJ) and 918 / 1211 (OBJ) and much fewer than A0 and A1 labels. Thus, SR labels can connect much more arguments to their predicates.

### 3 Approach

This section describes our new approach to sentence compression. In order to introduce rich syntactic and semantic constraints to a sentence compression model, we employ Markov Logic (Richardson and Domingos, 2006). Since Markov Logic supports both soft and hard constraints, we can implement our SR constraints in simple and direct fashion. Moreover, implementations of learning and inference methods are already provided in existing Markov Logic interpreters such as *Alchemy* ² and *Markov thebeast*. ³ Thus, we can focus our effort

²[http://alchemy.cs.washington.edu/](http://alchemy.cs.washington.edu/)

³[http://code.google.com/p/thebeast/](http://code.google.com/p/thebeast/)

Sense information is too sparse because the size of the WNC Corpus is not big enough.
Table 2: MLN Predicates

| predicate | definition |
|-----------|------------|
| inComp(i) | Token i is in compression |
| pred(i)  | Token i is a predicate |
| role(i, j, r) | Token i has an argument j with role r |
| word(i, w) | Token i has word w |
| pos(i, p) | Token i has Pos tag p |
| dep(i, j, d) | Token i is dependent on token j with dependency label d |
| path(i, j, l) | Tokens i and j has syntactic path l |
| height(i, n) | Token i has height n in dependency tree |

Table 2: MLN Predicates

role r. These formulae are called global formulae because they have more than two hidden MLN predicates. With global formulae, our model makes two decisions at a time. When considering the example in Figure 1, Formula (3) will be grounded as:

\[
\text{role}(9, 1, A0) \land \text{inComp}(9) \Rightarrow \text{inComp}(1) \quad (5)
\]

\[
\text{role}(9, 7, \text{AM-TMP}) \land \text{inComp}(9) \Rightarrow \text{inComp}(7). \quad (6)
\]

In fact, Formula (5) gains a higher weight than Formula (6) by learning on training data. As a result, our system gives “1-Harari” more chance to survive in compression. We also add some extensions of Formula (3) combined with \(\text{dep}(i, j, +d)\) and \(\text{path}(i, j, +l)\) which enhance SR constraints. Note, all our SR constraints are “predicate-driven” (only \(\Rightarrow\) not \(\Leftrightarrow\) as in Formula (13)). Because an argument is usually related to multiple predicates, it is difficult to model “argument-driven” formula.

3.1.2 Lexical and Syntactic Features

For lexical and syntactic features, we mainly refer to the previous work (McDonald, 2006; Clarke and Lapata, 2008). The first two formulae in this section capture the relation of the tokens with their lexical and syntactic properties. The formula describing such a local property of a word form is

\[
\text{word}(i, +w) \Rightarrow \text{inComp}(i) \quad (7)
\]

which implies that a token i is in compression with a weight that depends on the word form.

For part-of-speech (POS), we add unigram and bigram features with the formulae,

\[
\text{pos}(i, +p) \Rightarrow \text{inComp}(i) \quad (8)
\]

\[
\text{pos}(i, +p_1) \land \text{pos}(i + 1, +p_2) \Rightarrow \text{inComp}(i). \quad (9)
\]

POS features are often more reasonable than word form features to combine with the other properties. The formula,

\[
\text{pos}(i, +p) \land \text{height}(i, +n) \Rightarrow \text{inComp}(i). \quad (10)
\]

is a combination of POS features and a height in a dependency tree.

The next formula combines POS bigram features with dependency relations.

\[
\text{pos}(i, +p_1) \land \text{pos}(j, +p_2) \land \text{dep}(i, j, +d) \Rightarrow \text{inComp}(i). \quad (11)
\]

Moreover, our model includes the following global formulae,

\[
\text{dep}(i, j, +d) \land \text{inComp}(i) \Rightarrow \text{inComp}(j) \quad (12)
\]

\[
\text{dep}(i, j, +d) \land \text{inComp}(i) \Leftrightarrow \text{inComp}(j) \quad (13)
\]

which enforce the consistencies between head and modifier tokens. Formula (12) represents that if we include a head token in compression then its modifier must also be included. Formula (13) ensures that head and modifier words must be simultaneously kept in compression or dropped. Though Clarke and Lapata (2008) implemented these dependency constraints by ILP, we implement them by soft constraints of MLN. Note that Formula (12) expresses the same properties as Formula (3) replacing \(\text{dep}(i, j, +d)\) by \(\text{role}(i, j, +r)\).

4 Experiment and Result

4.1 Experimental Setup

Our experimental setting follows previous work (Clarke and Lapata, 2008). As stated in Section 2, we employed the WNC Corpus. For preprocessing, we performed POS tagging by stanford-tagger. 5 and dependency parsing by MST-parser (McDonald et al., 2005). In addition, LTH 6 was exploited to perform both dependency parsing and SR labeling. We implemented our model by Markov Thebeast with Gurobi optimizer. 7

Our evaluation consists of two types of automatic evaluations. The first evaluation is dependency based evaluation same as Riezler et al. (2003). We performed dependency parsing on gold data and system outputs by RASP. 8 Then we calculated precision, recall, and F1 for the set of label(head, modifier).

In order to demonstrate how well our SR constraints keep correct predicate-argument structures in compression, we propose SRL based evaluation. We performed SRL labeling on gold data.
As future work, we will compare our model with the other state-of-the-art systems. We will also investigate the correlation between readability and SRL-based score by manual evaluations. Furthermore, we would like to combine discourse constraints with SR constraints.
References
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.

James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics.

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

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT ’10, pages 885–893, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187. Association for Computational Linguistics.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 716–724, Beijing, China, August. Coling 2010 Organizing Committee.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT ’05, pages 523–530, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*, pages 297–304.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 118–125. Association for Computational Linguistics.