Parallel Sentence Compression

Julia Ive, François Yvon

Parallel Sentence Compression

Julia Ive[1,2], **François Yvon**[1]
LIMSI, CNRS, Univ Paris-Sud, Université Paris-Saclay, 91 403 Orsay, France,[1]
Cochrane France, INSERM U1153, 75181 Paris, France[2]
{firstname.lastname}@limsi.fr

Abstract

Sentence compression is a way to perform text simplification and is usually handled in a monolingual setting. In this paper, we study ways to extend sentence compression in a bilingual context, where the goal is to obtain parallel compressions of parallel sentences. This can be beneficial for a series of multilingual natural language processing (NLP) tasks. We compare two ways to take bilingual information into account when compressing parallel sentences. Their efficiency is contrasted on a parallel corpus of News articles.

1 Introduction

Text simplification is a well studied application of Natural Language Processing (NLP) techniques. Its main goal is to reduce the complexity of a text without degrading the informational content. This task proves useful for a wide range of applications, be they human-oriented (e.g. text adaptation for language learning purposes, for people with reading disabilities etc. (Siddharthan, 2014; Klaper et al., 2013)) or machine-oriented, serving as a basis to improve the efficiency of other NLP downstream components (e.g., parsing (Jonnalagadda et al., 2009), semantic role labeling (Vickrey and Koller, 2008) etc.). Simplification can be performed at different linguistic levels: lexical (Paetzold and Specia, 2016), syntactic (Siddharthan, 2011), or both (Paetzold and Specia, 2013).

Sentence compression is a way to perform simplification at the level of sentences, by reducing the sentence length without sacrificing important information. Many works only consider purely syntactic simplifications, though lexical changes are also possible, especially in language learning scenarios (Cohn and Lapata, 2008; Napoles et al., 2011a).

By and large, the motivations that have been put forward for monolingual sentence compression can be also used to motivate bilingual sentence compression, understood here as the generation of parallel compressions of parallel sentences. Bilingual Sentence Compression can be used, for instance, to produce simpler versions of a parallel text for learning purposes, or to generate summaries and subtitles in different languages, or even to build simplified parallel corpora for training a Machine Translation (MT) system.

Parallel sentence compression can be approached in many ways: it is first possible to compress independently each side of a bitext using monolingual simplification, an approach which however runs the risk of breaking the parallelism of the resulting corpus. Compression could also be performed in a symmetric manner by generalizing monolingual algorithms to the case of parallel sentences. We focus here on an **asymmetrical scenario**, where the target compression is a translation of a previously simplified source sentence.

In this context, our main contribution consists in studying various bitext compression methodologies that should **ensure the parallelism of the simplified bitext**, which, to our knowledge, is the first attempt of this kind. Two compression methods are developed and compared in Section 2 : (1) a dynamic programming (DP) approach, considering the final compression

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1503–1513, Osaka, Japan, December 11-17 2016.

as a result of a series of local optimal decisions and (2) an integer linear programming (ILP) method, capable to handle global constraints. These methods are used to compress texts in the News domain (see Section 3); we experiment there with data distributed in the context of the WMT'15 translation task (English-French, automatic compression attempted on French).

2 Bilingual and Monolingual Methods for Sentence Compression

In our asymmetrical bilingual context, we formulate the compression problem as follows: given a source sentence $\mathbf{e} = e_1, e_2, \ldots, e_J$, its compressed version $\mathbf{e}' = e'_1, e'_2, \ldots, e'_{J'}$ and its translation $\mathbf{f} = f_1, f_2, \ldots, f_I$, we search for $\mathbf{f}' = f'_1, f'_2, \ldots, f'_{I'}$ that translates $\mathbf{e}'$. $\mathbf{f}'$ should both **preserve the meaning** of $\mathbf{e}'$ and respect the **grammaticality** requirements of the target language. Most approaches to monolingual compression further assume that a (dependency) parse tree of $\mathbf{f}$ is available, taking the form of a set of (dependent, head) pairs. We make the same assumption here, denoting $\tau = \{(f_i, h(f_i)), 1 \leq i \leq I\}$ this dependency tree.

Recent proposals for solving this task use dynamic programming (DP) techniques (McDonald, 2006; Filippova, 2010) or integer linear programming (ILP) (Clarke and Lapata, 2008; Filippova et al., 2015), in both cases actively taking syntactic information into account. Inspired by those approaches, we propose below two methods for bilingual compression, enriched with MT-related bilingual information. We also include a description of our baseline compression system, based on two independent monolingual compressions.

2.1 Compressing with Finite-State Machines and Dynamic Programming (DPbi)

Our first approach to compression (DPBI) uses finite-state techniques. Recall that a weighted finite-state automaton (WFSA) over a set of weights $\mathbb{K}$ is represented by a 7-tuple $A = (\Sigma, Q, B, F, E, \lambda, \rho)$, where Σ is a finite alphabet, Q is a finite set of states, $B \subseteq Q$ contains the initial states and $F \subseteq Q$ the final states; $E \subseteq Q \times \Sigma \times \mathbb{K} \times Q$ is a set of weighted transitions, $\lambda : B \to \mathbb{K}$ and $\rho : F \to \mathbb{K}$ are respectively the initial and final weight functions (Mohri, 1997).

Given $\mathbf{f}$, the search space for compression is built as follows: assuming $\mathbf{f}$ conventionally starts (respectively ends) with `<s>` at index f_0 (resp. `</s>` at index f_{I+1}), we first build the standard automaton $A_{\mathbf{f}}$ for $\mathbf{f}$, the states of which correspond to the prefixes of $\mathbf{f}$.

We then add "skip" transitions $(q_k, f_l, w, q_l,), \forall l > k+1$. In this step, we make sure to preserve the syntactic dependency relationships so as to ensure that the subgraph induced by words in the compression is a subtree of the complete dependency tree. To this end, skip transitions $(q_k, f_l, w, q_l,)$ are created subject to the condition that $\forall m, k < m < l$, f_m is neither an ancestor of f_k nor an ancestor of f_l.

When the dependency trees are projective, these conditions are sufficient to ensure that compressions will be grammatical: if a compression contains a word f_i, it will also contain its head. To see why, assume with no loss of generality that $h(f_i) < f_i$ for some i. If f_i is in the compression, the incoming arc in $A_{\mathbf{f}}$ either starts in node f_k with $k < l$, or $h(f_i)$ precedes f_k (it cannot be skipped). Either $f_k = h(f_i)$, which is what we seek; or f_k is a descendant of $h(f_i)$. We can then repeat the same argument with the arc labeled f_k. It is also routine to check that all possible grammatical compressions can be obtained in this way.

Transitions $(q_k, f_l, w, q_l,)$, with $l > k + 1$, are weighted according to a score w aggregating:

- S_{LM} - a 2-gram language model (LM) score : $S_{LM}(f_l|f_k) = \log P(f_l|f_k)$. The generalization to higher-order n-grams is straightforward. In our implementation, we have used a POS-based LM, as we believe it will provide a better generalization than a word-based LM;

- $S_{IBM}(f_l|\mathbf{e}')$ - the posterior log-probability of f_l in the IBM model 1 of Brown et al. (1993):

$$S_{IBM}(f_l|\mathbf{e}') = \log \big(\frac{1}{(J'+1)} \sum_{j=1}^{J'} t(f_l|e'_j)\big). \quad (1)$$

These scores are summed along a path and yield the total IBM score: $IBM_1(\mathbf{f}'|\mathbf{e}')$.

- $S_{ali}(f_l)$ - this score approximates the contribution of f_l to the posterior log-probability of the $IBM_1(\mathbf{e}'|\mathbf{f})$ at the sentence level:

$$S_{ali}(f_l) = \sum_{i=1}^{J'} \mathbb{1}\{f_l = \operatorname*{argmax}_f t(e'_i|f)\} t(e'_i|f_l), \tag{2}$$

  where $\mathbb{1}\{T\}$ is the indicator function for predicate T. This approximation is required due to the impossibility to decompose the inverse IBM model 1 score over the arcs of $A_{\mathbf{f}}$.

The use of IBM model 1 scores is meant to ensure the preservation of the meaning of the provided source compression $\mathbf{e}'$, hence the parallelism of the resulting sentence pair. Each arc is additionally weighted with a word penalty, which should ensure that short paths are not improperly given preference over longer ones.

The score of a path generating a target string $\mathbf{f}'$ is finally computed as a summation of all arcs in the path:

$$S(\mathbf{f}'|\mathbf{e}') = \alpha \cdot S_{LM}(\mathbf{f}') + \beta \cdot S_{IBM}(\mathbf{f}'|\mathbf{e}') + \gamma \cdot S_{ali}(\mathbf{f}') + \delta \cdot l_{\mathbf{f}'}, \tag{3}$$

where α, β, γ and δ are tunable parameters. The optimal target compression is computed via standard shortest path techniques. Note that this approach can be generalized in many ways, notably including additional costs, subject to *locality constraints*: scoring functions evaluating properties of the compressed sentence should decompose over the arcs of the automaton. This restriction warrants a more generic approach to the problem, relying on Integer Linear Programming.

2.2 Compressing with Integer Linear Programming (ILPbi)

Integer Linear Programming (ILP) (Dantzig and Thapa, 1997) is an optimization approach to solving combinatorial problems that can be expressed as linear programs and in which some or all of the variables are restricted to be non-negative integers. In the following, we heavily rely on the work of Clarke and Lapata (2008), who develop an approach based on ILP for monolingual sentence compression.

The formulation of an ILP problem requires the definition of:

- decision variables, they will be binary in our case;
- an objective function, corresponding to the compression score we wish to maximize;
- constraints, i.e. linguistic or consistency conditions restricting the possible values of decision variables.

The main decision variables in our problem indicate whether a target word f_i initially in $\mathbf{f}$ also occurs in the compressed text:

$$\forall i \in [1 \dots I], x_i = \begin{cases} 1 & \text{if } f_i \text{ is in the compression} \\ 0 & \text{otherwise} \end{cases}$$

Following again Clarke and Lapata (2008), we define additional decision variables for the 2-gram LM scores (again, the generalization to higher-order n-grams is straightforward):

$$\forall i \in [1 \dots I], y_i = \begin{cases} 1 & \text{if } f_i \text{ starts the compression} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall i \in [1 \dots I], p_i = \begin{cases} 1 & \text{if } f_i \text{ ends the compression} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall i \in [1 \dots I-1], \forall k \in [i+1 \dots I], z_{ik} = \begin{cases} 1 & \text{if the pair } (f_i, f_k) \text{ is in the compression} \\ 0 & \text{otherwise} \end{cases}$$

Our ILPBI model is again integrated with IBM model 1 scores, which help ensure the preservation of meaning in $\mathbf{e}'$. For our approximation of the $IBM_1(\mathbf{e}'|\mathbf{f})$ score described in Section 2.1, we introduce a new variable:

$$\forall i \in [1 \dots I], \forall m \in [1 \dots J], a_{im} = \begin{cases} 1 & \text{if } g(t(\mathbf{e}'_m|f_i)) > 0 \\ 0 & \text{if } g(t(\mathbf{e}'_m|f_i)) = 0 \end{cases}, \tag{4}$$

where $g(t(\mathbf{e}'_m|f_i)) = \mathbb{1}_{f=\text{argmax}_f\, t(\mathbf{e}'_m|\mathbf{f})}(t(\mathbf{e}'_m|f_i))$ tests whether f_i is the best alignment for $\mathbf{e}'_m$.

Our objective function models 2-gram LM scores, as well as bilingual $IBM_1(\mathbf{f}|\mathbf{e}')$ and our approximation of $IBM_1(\mathbf{e}'|\mathbf{f})$:

$$\begin{aligned} S(\mathbf{f}'|\mathbf{e}') \; =& \alpha \sum_{i=1}^{I} x_i \cdot \log\big(\frac{1}{(I+1)} \sum_{i=1}^{I} t(f_l|e'_i)\big) + \gamma \sum_{i=1}^{I} y_i \cdot \log P(f_i|\text{<s>}) \\ &+ \gamma \sum_{i=1}^{I-1} \sum_{k=i+1}^{I} z_{ik} \cdot \log P(f_k|f_i) + \beta \sum_{m=1}^{J} \sum_{i=1}^{I} a_{i,m} \cdot \log t(\mathbf{e}'_m \mathbf{f}_i) + \gamma \sum_{i=1}^{I} p_i \cdot \log P(\text{</s>}|f_i) \end{aligned} \tag{5}$$

subject to: $x_i, y_i, z_{ik}, p_i, a_{i,m} \in \{0, 1\}$. The following constraints are also applied for generation of valid n-gram sequences without word repetition:

Constraint 1 Exactly one word can start a compression.

$$\sum_{i=1}^{I} y_i = 1 \tag{6}$$

Constraint 2 If a word is in the compression it must either start it, or must follow another word.

$$\forall k : k \in [1 \dots I], x_k - y_k - \sum_{i=1}^{k-1} z_{ik} = 0 \tag{7}$$

Constraint 3 If a word is in the compression it must either be followed by another word or end the sentence.

$$\forall i : i \in [1 \dots I], x_i - (\sum_{k=i+1}^{I} z_{ik}) - p_i = 0 \tag{8}$$

Constraint 4 Exactly one word can end a compression.

$$\sum_{i=1}^{I} p_i = 1 \tag{9}$$

Constraint 5 A dependent f_i cannot be included in a compression without its head $h(f_i)$ from the dependency tree $\tau_{\mathbf{f}}$, the constraint that ensures the grammaticality of $\mathbf{f}'$. This is a simplified version of constraints (20)-(24) of (Clarke and Lapata, 2008).

$$\forall i, x_{h(i)} - x_i \geq 0 \tag{10}$$

Constraint 6 If a compression contains a left bracket/quote mark it should contain the right bracket/quote mark.

$$x_{left} - x_{right} = 0 \tag{11}$$

Constraint 7 A compression is to be at least b tokens long. This constraint controls the compression length and prevents the model to generate too short compression preferred by LM.

$$\sum_{i=1}^{I} x_i \geq b \tag{12}$$

Constraint 8 If a word is in the compression it can have a best alignment in $\mathbf{e}'$.

$$x_i - a_{im} >= 0 \tag{13}$$

To complete this section, we now present our tools for monolingual compression, that will be used in our baseline system.

2.3 Monolingual Compression (DPmono and ILPmono)

In the monolingual scenario, we are given a target sentence $\mathbf{f} = f_1, f_2, \ldots, f_I$. The goal is to produce a target compression $\mathbf{f}' = f'_1, f'_2, \ldots, f'_{I'}$ by removing any subset of words in $\mathbf{f}$, given the dependency tree τ. Here we also assume to be given the lemmas corresponding to the words in $\mathbf{f}$: $m(f)$ denotes the lemma associated with word f. Here again, we look for $\mathbf{f}'$ that should be grammatical and preserve the main aspects of the meaning of $\mathbf{f}$.

We introduce grammaticality constraints in our compressions in a way similar to our bilingual methods. To help meaning preservation we introduce the following semantic importance score for the sense-bearing words of $\mathbf{f}$, namely nouns, verbs, adjectives and adverbs, inspired again by Clarke and Lapata (2008):

$$S_{sem} = \frac{1}{D_{f_{sb}}} fr_{m(sb)} \log \frac{F}{F_{m(sb)}} \tag{14}$$

This score is a `TD-IDF` measure for the lemma $m(f)$ of each sense-bearing word, weighted proportionally to its depth in the dependency tree $D_{f_{sb}}$, where $fr_{m(sb)}$ and $F_{m(sb)}$ are respectively the frequency of the lemma in the news article and in the corpus, and F is the count of all sense-bearing lemmas in the corpus.

1. DPmono For the DPmono approach, we construct an WFSA in a way similar to the one described in 2.1. We weight each arc in the automaton with S_{LM} and S_{sem}. The score of a path generating a target string $\mathbf{f}'$ is computed as:

$$S(\mathbf{f}'|\mathbf{e}') = \alpha \cdot S_{LM}(\mathbf{f}') + \beta \cdot S_{sem}(\mathbf{f}') + \gamma \cdot l_{\mathbf{f}'}, \tag{15}$$

where α, β and γ are tunable parameters.

2. ILPmono The objective function for ILPmono similarly models $S_{sem}(f_i)$ and 2-gram LM scores:

$$S(\mathbf{f}'|\mathbf{e}') = \sum_{i=1}^{I} x_i \cdot \theta S_{sem}(f_i) + \eta \sum_{i=1}^{I} y_i \cdot \log P(f_i|\text{<s>}) + \eta \sum_{i=1}^{I-1} \sum_{k=i+1}^{I} z_{ik} \cdot \log P(f_k|f_i) + \eta \sum_{i=1}^{I} p_i \cdot \log P(\text{</s>}|f_i),$$

subject to: $x_i, y_i, z_{ik}, p_i \in \{0, 1\}$, with θ and η as tunable parameters. Constraints 1-7 of ILPbi are applied.

3 Experimental Setup

In our evaluation experiments we were guided by the following questions: (1) Does the proposed bilingual compression methodology ensure **the resulting corpus parallelism** (compressed target text succeeds in preserving the meaning of the input compressed source text)? (2) Which of the proposed bilingual compression methods is more **efficient**?

3.1 Metrics

To answer the above questions, we compared the results of our bilingual methods DPbi and ILPbi to the results of our monolingual methods DPmono and ILPmono (baselines) using the compression rate (CompR) estimation, F-score metric for the relations in grammatical dependency parse trees (Chen and Manning, 2014), as well as the standard MT metric BLEU (Papineni et al., 2002; Clarke and Lapata, 2008; Napoles et al., 2011b). F-score measures how much meaning is preserved in the compression using the grammatical-functional information, BLEU measures the fluency of the produced compressions.

We also computed the confidence score (CS) of the parallelism between the produced compressions and the compressed source. We used a Logistic Regression model trained on the parallel sentences extracted from a corpus of manual alignments. The model exploits such features as the length difference ratio, IBM model 1 scores, the cosine similarity of the distributional representations for source and target etc. Details regarding the model are in (Xu et al., 2015).

3.2 Data and Translation Models

Our experiments used the News data provided by the organizers of the WMT'15 news translation task (English-French) [1]. Details regarding data preparation are in (Marie et al., 2015).

We randomly chose around 60 articles from the News 2014 test set to be used as our development and test sets. The compressed source and reference data were created manually by three annotators. Two of them were native French speakers, all the three annotators were fluent in both English and French. The annotators were given the instruction to delete any quantity of words without disturbing the meaning and the grammaticality of a source English sentence. Target French words "translating" deleted source words were deleted accordingly under the condition that a sentence stays grammatical. This resulted in the small compression rate of CompR = 79.74 at the source side and of CompR = 84.45 at the target side [2].

In our experiments, we used the French 3-gram POS LMs trained with Witten-Bell smoothing on a part of the target side of the parallel Giga corpus (≈14% of the available corpus) using the SRILM (Stolcke, 2002) toolkit. The same parallel data and the uncompressed development and test sets were used to compute the IBM1 scores in both directions with the help of MGIZA++ (Gao and Vogel, 2008), and to estimate the lemma frequencies for the semantic importance scores (Schmid, 1995). The dependency parse trees were produced using the Stanford parser (Chen and Manning, 2014).

The statistics of the development and test data, as well as the data used for model estimation are in Table 1.

		Full		Compression	
	lines	tok., en	tok., fr	tok., en	tok., fr
development set	504	11K	13K	9K	11K
test set	489	12K	14K	10K	12K
Giga	3M	71M	86M		

Table 1: Data Statistics

For the WFSA experiments, we mostly used the SriLM toolkit (Stolcke, 2002). The ILP

[1] http://www.statmt.org/wmt15/translation-task.html

[2] The traces of compression operations for the corpus are available at http://perso.limsi.fr/ive.

experiments were performed with the GLPK toolkit.[3] All the tunable parameters were tuned to optimize BLEU. In our implementation of ILPBI, we slightly modified constraint 5 to take into account the negation relations: when the head of such relation are included in the compression, then their immediate dependent must also be selected. This is because the preservation of such relation is crucial for the overall meaning of a sentence. Finally, constraint 7 is parametrized by a preset compression ratio, rather than a preset target length; in our experiments this compression ratio was set to match the compression ratio of DPBI so as to make our results more comparable.

3.3 Evaluation

Results of our experiments comparing bilingual and monolingual methods are in Table 2.

model	CompR	F-score	BLEU	CS
DPmono	82.58	80.37	73.37	0.94
DPBI	84.63	81.71	**76.09**	0.96
ILPmono	85.10	80.18	62.59	0.95
ILPBI	85.06	**82.72**	69.85	**0.97**
Ref.	83.59			0.99

Table 2: Evaluation results

As reflected by the automatic metrics, the bilingual methods produce compressions that better preserve the meaning of the source sentence than the corresponding monolingual methods, hence improving the parallelism of the resulting compressions (average ΔF-score = 1.94, ΔBLEU = 4.99 and ΔCS = 0.02, between the ILP and DP monolingual and bilingual methods) (see Table 3).

Ref.	Irak: **octobre** a été le mois le plus sanglant depuis 2008
	'Iraq: **October** was the bloodiest month since 2008'
DPmono	Irak: a été le mois le plus sanglant depuis 2008
	'Iraq: was the bloodiest month since 2008'
DPBI	Irak : **octobre** a été le mois le plus sanglant depuis 2008
	'Iraq: **October** was the bloodiest month since 2008'
Ref.	La ville de New York en envisage **un**.
	'The city of New York is considering **one**.'
ILPmono	La ville de New York en envisage.
	'The city of New York is considering.'
ILPBI	Ville de New York en envisage **un**.
	'City of New York is considering **one**.'

Table 3: Examples of compressions produced by the monolingual and bilingual methods

F-score and CS estimations suggest that ILPBI is a slightly more efficient method than DPBI in terms of preserving parallelism (+1.01 F-score, +0.01 CS). Due to the global constraints, ILPBI tends to include more sense-bearing words in the compression (see Table 4, first example). Our example shows that ILPBI kept the word "cour" 'court'. This noun is more important for understanding the meaning of the sentence than the preposition "en" 'in' following the verb "présenter" 'appear', chosen due to the local decision taken by DPBI.

At the same time the length constraint "obliges" ILPBI to compress every sentence. In our setting with the small compression rate, short sentences often stay uncompressed. In this case, DPBI is able to choose the automaton path of the maximum length (see Table 4, second example). ILPBI though in this case deletes the auxiliary words (articles in our example). This

[3] `http://glpk-java.sourceforge.net`

is reflected in the decrease of the BLEU score for ILPBI as compared to DPBI (-6.24 BLEU). BLEU here is heavily penalized by the decrease in matching n-grams of the order $n > 1$.

ILP methods can also be very convenient for a series of reasons. E.g., in the absence of development corpora ILP compressions can be obtained with minimum or without any tuning. Another advantage is the easy parameterization of the compression rate. This criteria was very important for our task with the small compression rate. Keeping it was crucial for correct evaluation of our methodology. Thus, for the DP methods we observe the compression rate variation of ΔCompR = 2.05, for the ILP methods this variation is insignificant (ΔCompR = 0.04).

Ref.	Omar Hassan est toujours en détention et se présenter **en cour** vendredi. 'Omar is still in custody and will appear **in court** on Friday.'
ILPBI	Omar Hassan est toujours en détention et se présenter **cour** vendredi. 'Omar is still in custody and will appear **court** on Friday.'
DPBI	Omar Hassan est toujours en détention et se présenter **en** vendredi. 'Omar is still in custody and will appear **in** on Friday.'
Ref.	Après **un** accord de paix signé en 1992, elle est devenue **un** parti d'opposition . 'After **a** peace agreement signed in 1992, it became **an** opposition party.'
ILPBI	Après accord de paix signé en 1992, elle est devenue parti opposition. 'After peace agreement signed in 1992, it became opposition party.'
DPBI	Après **un** accord de paix signé en 1992, elle est devenue **un** parti d'opposition . 'After **a** peace agreement signed in 1992, it became **an** opposition party.'

Table 4: Examples of ILPBI and DPBI compressions

4 Related Work

The deletion-based compression problem has been studied using a series of modeling paradigms. We mention first the work of Knight and Marcu (2002), who use the *noisy channel* model. This approach aims to maximize $P(\mathbf{f}'|\mathbf{f}) \propto P(\mathbf{f}')P(\mathbf{f}|\mathbf{f}')$, where $P(\mathbf{f}')$ is the source model, and $P(\mathbf{f}|\mathbf{f}')$ models the syntactic parse tree probability of the long sentence being an expansion of the compressed one. The noisy channel model is also used by approaches that consider compression as a monolingual translation problem (Napoles et al., 2016).

McDonald (2006) formulates the problem as a binary sequence labeling problem with a rich syntactic feature set, and proposes a solving procedure based on dynamic programming techniques. More recent DP solutions to the sentence compression problem use neural network architectures (Filippova et al., 2015).

The ILP approach to compression was introduced by Clarke and Lapata (2008). The main motivation was the necessity to take global features into account (e.g., the constraint to have at least one verb in the compressed sentences). This approach has been widely reused in research related to text compression with various modifications to syntactic and informativeness scores used by Clarke and Lapata (2008) (see also (Wei et al., 2015; Filippova and Altun, 2013)).

In our bilingual framework we compare the performance of DP and ILP approaches. As far as we know this is the first attempt to create compressed parallel bitext in asymmetrical setting. A closely related work is that of Aziz et al. (2012), who also exploits bilingual information. The authors propose a PBSMT solution for joint translation and compression of subtitles, which dynamically decides where it is necessary to impose a space/time constraint on the translated text.

5 Conclusions

In this paper we consider sentence compression in a bilingual setting. We adopt an asymmetrical view to the task, where we first compress the source, then look for a compressed target translating the reduced source. Based on recent research on these issues, our main contribution is to adapt

existing monolingual compression techniques to produce compressed bitext. As we know this the first attempt of the kind. We use dynamic programming (DP) and integer linear programming methods (ILP) enriched with bilingual features. Both methods improve the preservation of the compressed source meaning, hence the parallelism of the resulting bitext, as opposed to using independently monolingual methods in source and target.

In our setting, ILP was found to perform better than DP; the ILP method is more flexible and requires less resources for tuning; furthermore, it can accommodate more complex (e.g. global) constraints. Our future work includes exploring additional global constraints, experimenting with shorter compressions, as well as using basic phrase-based statistical machine translation (PBSMT) techniques, including applying the noisy channel model and beam-search decoding to find the best possible target compression.

The results of bilingual compression can be used for human-oriented (language learning, subtitles generation etc.) purposes, or in a large spectrum of natural language processing (NLP) tasks. The approach can be extended to paraphrastic compression (as opposed to deletion-based), as well as applied in the symmetric compression scenario, when source and target are compressed simultaneously.

Acknowledgements

The work of the first author is supported by a CIFRE grant from the French ANRT. We would like to thank Yong Xu for helping us with the computation of the parallelism confidence scores, as well as the annotators for their participation in the creation of the compressed corpus.

References

Wilker Aziz, Sheila Castilho Monteiro de Sousa, and Lucia Specia. 2012. Cross-lingual sentence compression for subtitles. In *16th Annual Conference of the European Association for Machine Translation*, EAMT, pages 103–110, Trento, Italy.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression an integer linear programming approach. *J. Artif. Int. Res.*, 31(1):399–429, March.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK, August. Coling 2008 Organizing Committee.

George B. Dantzig and Mukund N. Thapa. 1997. *Linear Programming 1: Introduction.* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP '13)*, pages 1481–1491. (Click on the Abstract link to get access to the described dataset).

Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15).*

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 322–330, Beijing, China, August. Coling 2010 Organizing Committee.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June. Association for Computational Linguistics.

Siddhartha Jonnalagadda, Luis Tari, Jörg Hakenberg, Chitta Baral, and Graciela Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. 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 177–180, Boulder, Colorado, June. Association for Computational Linguistics.

David Klaper, Sarah Ebling, and Martin Volk. 2013. Building a german/simple german parallel corpus for automatic text simplification. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 11–19, Sofia, Bulgaria, August. Association for Computational Linguistics.

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

Benjamin Marie, Alexandre Allauzen, Franck Burlot, Quoc-Khanh Do, Julia Ive, Elena Knyazeva, Matthieu Labeau, Thomas Lavergne, Kevin Löser, Nicolas Pécheux, and François Yvon. 2015. LIMSI@WMT'15 : Translation task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 145–151, Lisbon, Portugal, September. Association for Computational Linguistics.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 297–304.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:269–311.

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011a. 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, Portland, Oregon, June. Association for Computational Linguistics.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011b. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 91–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Courtney Napoles, Chris Callison-Burch, and Matt Post. 2016. Sentential paraphrasing as black-box machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 62–66, San Diego, California, June. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2013. Text simplification as tree transduction. In *Ninth Brazilian Symposium in Information and Human Language Technology*, STIL, pages 116–125, Fortaleza, Brazil.

Gustavo Paetzold and Lucia Specia. 2016. Benchmarking lexical simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, Philadelphia, US.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Advaith Siddharthan. 2011. Text simplification using typed dependencies: A comparision of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 2–11, Nancy, France, September. Association for Computational Linguistics.

Advaith Siddharthan. 2014. A survey of research on text simplification. *International Journal of Applied Linguistics*, 165(2):259–298.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, Denver, Colorado, September.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio, June. Association for Computational Linguistics.

Zhongyu Wei, Yang Liu, Chen Li, and Wei Gao. 2015. Using tweets to help sentence compression for news highlights generation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 50–56, Beijing, China, July. Association for Computational Linguistics.

Yong Xu, Aurélien Max, and François Yvon. 2015. Sentence alignment for literary texts. *Linguistic Issues in Language Technology*, 12(6):25 pages.