Lexical Chains meet Word Embeddings in Document-level Statistical Machine Translation

Laura Mascarell
Institute of Computational Linguistics, University of Zürich
mascarell@cl.uzh.ch

Abstract

The phrase-based Statistical Machine Translation (SMT) approach deals with sentences in isolation, making it difficult to consider discourse context in translation. This poses a challenge for ambiguous words that need discourse knowledge to be correctly translated. We propose a method that benefits from the semantic similarity in lexical chains to improve SMT output by integrating it in a document-level decoder. We focus on word embeddings to deal with the lexical chains, contrary to the traditional approach that uses lexical resources. Experimental results on German→English show that our method produces correct translations in up to 88% of the changes, improving the translation in 36%-48% of them over the baseline.

1 Introduction

Current phrase-based Statistical Machine Translation (SMT) systems translate sentences in a document independently (Koehn et al., 2003), ignoring document context. This sentence-level approach causes wrong translations when discourse knowledge is needed. Therefore, many methods that integrate discourse features have been proposed to improve lexical choice.

Documents are a set of sentences that function as a unit. When we translate at document-level we take into account document properties that help to improve the quality of the translation, not only locally, but also in the context of the document. Coherence and cohesion are terms that describe properties of texts. Coherence concerns the semantic meaningfulness of the text, whereas cohesion has to do with relating the sentences through reference, ellipsis, substitution, conjunction, and the use of semantically-similar words. Often, these words are related sequentially in the document, defining the topic of the text segment that they cover. These sequences of words are lexical chains, and they have been successfully used in research areas such as information retrieval (Stairmand, 1996; Rinaldi, 2009) and document summarization (Barzilay and Elhadad, 1997; Pourvali and Abadeh, 2012). However, they have received little attention in Machine Translation (MT).

Galley and McKeown (2003) introduce a method to detect lexical chains using WordNet (Miller, 1995). The method first builds a representation of all words in the document and all their senses, creating semantic links such as synonym, hypernym, hyponym, and sibling between them. It then uses the semantic links to disambiguate each word and builds the lexical chains accordingly.

The performance of the method is evaluated on a sense disambiguation task. Indeed, lexical chains help to disambiguate the sense of polysemic words by looking at the words in the chain. Despite the problems of word senses (Kilgarriff, 1997, 2006; Hanks, 2000), it shows the potential that lexical chains have to improve the lexical choice of words with multiple translations in MT.

In this paper, we present a method that uses word embeddings instead of lexical resources to detect the lexical chains in the source and also to maintain their semantic similarity on the target side. We focus on the German→English translation and integrate our model into the document-level SMT decoder Docent (Hardmeier et al., 2013). We perform a manual evaluation of the output, which shows that our method improves the translation over the baseline, with a tendency to consistently translate the words in the chain. Furthermore, experimental results reveal that the use of word embeddings in lexical chain detection outperforms lexical resources on the translation task.

2 Related Work

The one-sense-per-discourse hypothesis (Gale et al., 1992) is applied in MT, revealing lexical choice errors when words in a document are inconsistently translated (Carpuat, 2009). As a consequence, several approaches improve lexical choice by enforcing consistency throughout the document. Tiedemann (2010) and Gong et al. (2011) use cache-models for this purpose, and Xiao et al. (2011) apply a three-steps procedure that consist of identifying the ambiguous words in a document, obtaining a set of consistent translation for each of them, and generating a new translation of the document, where the identified words are translated consistently. Pu et al. (2017) also study consistency in translation and train classifiers on syntactic and semantic features to predict how to consistently translate pairs of nouns in a document. More specifically, Mascarell et al. (2014) and Pu et al. (2015) benefit from text dependencies to improve the translation of words that refer back to compounds.

Guillou (2013) analyses *when* (i.e. genre) and *where* (i.e. part-of-speech) lexical consistency is desirable. The results suggest that nouns should be encouraged to be translated consistently throughout the document, across all genres. Additionally, consistent translation of verbs and adjectives is beneficial for technical and public information documents, respectively.

Garcia et al. (2017) implement a feature for the document-level decoder Docent that uses word embeddings to translate repeated words consistently. The manual evaluation reveals that 60% of the time the output improves over the baseline and 20% of the time is equivalent or equal.

Word embeddings have also been proposed for Word Sense Disambiguation (WSD) (Iacobacci et al., 2016). Previously, other approaches were introduced to utilise embeddings for supervised (Zhong and Ng, 2010; Rothe and Schütze, 2015; Taghipour and Ng, 2015) and knowledge-based WSD (Chen et al., 2014).

Other approaches focus on including topic modelling and topic distributions for disambiguation (Hasler et al., 2014). Xiong and Zhang (2013) translate the coherence chain of the source document and use it to produce a coherent translation.

Xiong et al. (2013) are the first to explore the benefits of using lexical chains in MT. They introduce lexical chain based cohesion models in a hierarchical phrase-based SMT system (Chiang, 2005) trained on Chinese→English. To do so, they first use Galley and McKeown (2003)'s method to detect the lexical chains in the source and next generate the target lexical chains that are used by their cohesion models. To generate these target lexical chains, they train MaxEnt classifiers — one per unique source chain word — that predict the translation of each word given the previous and the next word in the chain and the immediate surrounding context. This machine learning approach results in limitations concerning chain words from the test set that are infrequent or even missing in the training data. Later, Xiong and Zhang (2014) integrate a sense-based translation model also using MaxEnt classifiers.

3 A Lexical Chain Model for SMT

This section describes the proposed method to improve the quality of translation in SMT utilising lexical chains. The method works as follows: it first detects the lexical chains in the source document (Section 3.1) and feeds them into the Lexical Chain Translation Model (LCTM), which is integrated into the document-level decoder Docent (Hardmeier et al., 2013). The model then gets their counterpart in the target through word alignment and computes the LCTM score that contributes to the overall translation score in the SMT system (Section 3.2). The reminder of this section describes the method in more detail.

3.1 Detecting Source Lexical Chains

Our automatic method to detect and build lexical chains from a document is inspired by the approach proposed by Morris and Hirst (1991). Their approach consists of manually detecting those lexical chains using a thesaurus to find the similarity between words. Our method implements the manual algorithm, detecting and building the lexical chains automatically.

Instead of using a thesaurus, we use word embeddings to compute the semantic similarity. Word embeddings are representations of words in a vector-space, which are commonly exploited to compute similarity between words (Mikolov et al., 2013) (See discussion in Section 3.3).

The method works as follows. It processes sentences in a given document sequentially. For each content word c (i.e. nouns, verbs, and adjectives) in every sentence, it checks whether c is semanti-

ihr nächstes *smartphone* wird zwei *betriebssysteme* beherrschen.
Die amerikaner rechnen für die zukunft mit einem *handy*, auf dem der *benutzer* durch drücken einer einzigen taste zwischen verschiedenen *betriebssystemen* *umschalten* kann.
Die vorgelegten Pläne sehen vielversprechend aus.

Lexical Chain$_1$: {*umschalten* (“switch”), *betriebssystemen* (“operating system”), *benutzer* (“user”), *betriebssysteme* (“operating system”)}
Lexical Chain$_2$: {*handy* (“cell phone”), *smartphone* (“smart phone”)}

Figure 1: Output of our lexical chain detection method on three sentences from newstest2010.

cally related to the previous content words *c'* in a span of five sentences, as suggested by Morris and Hirst (1991). If *c* and *c'* are semantically related, we proceed as follows:

- If *c* and *c'* do not belong to any chain, we create a new chain consisting of *c* and *c'*.
- If *c'* is in a chain *ch*$_i$, we append *c* to *ch*$_i$.
- If *c* and *c'* belong to two different chains, we then merge both chains.

The detected lexical chains preserve the semantic link between related content words, creating also one-transitive links. That is, c_i links to c_{i+l} by transitivity if c_i links to c_{i+k} and c_{i+k} to c_{i+l}, where *i*<*k*<*l* (Morris and Hirst, 1991).

Every link to a word in the lexical chain gives context to disambiguate the word itself. Therefore, the more links are created, the better. One-transitive links are safe to consider, because they are still semantically related as indicated by Morris and Hirst (1991), but further than that leads to errors. As an example, they point to the following lexical chain: {*cow*, *sheep*, *wool*, *scarf*, *boots*, *hat*, *snow*}. Here, we observe that while consecutive words in the chain like *wool* and *scarf* are semantically related, *cow* and *snow* are not. Figure 1 shows the lexical chains detected with our method on three sentences extracted from the document `idnes.cz/2009/12/11/76504` in newstest2010.[1]

3.2 The Lexical Chain Translation Model

In order to improve translation quality utilising lexical chains, we develop a model that favours document translations where the words in the target lexical chain are semantically related. The target lexical chains are the corresponding counterpart of the source lexical chains detected, and they are obtained by the LCTM through word alignment.

3.2.1 Integration into Docent

The LCTM is integrated as an additional feature function in the document-level decoder Docent as a standard SMT model:

$$f(s,t) = \sum_k \lambda_k h_k(s,t), \qquad (1)$$

where h_k are feature functions scores and λ_k their corresponding weight, obtained with the MERT optimisation technique (Och, 2003).

To understand how the model is integrated into Docent, we summarise how Docent works. Docent implements a search procedure based on local search. At every stage of the search, the decoder randomly applies a state operation such as `change-phrase-translation` (replaces the translation of a phrase with another from the phrase table), `swap-phrases` (exchanges phrases), `move-phrases` (randomly moves phrases in the sentence), and `resegment` (changes the segmentation of the source phrase). The search algorithm accepts then a new state (i.e. a new translation of the document), when its document score computed by Equation 1 is higher than the last accepted. To compute the document score, it considers the score obtained from each feature function. The initial translation of the whole document is either randomly generated or a translation from Moses (Koehn et al., 2007).

The LCTM is implemented as one of the feature functions in Docent, and therefore it contributes to the overall document score. Consider the example in Figure 2. This example shows two hypothetical Docent states when applying the state operation `change-phrase-translation` on the German word *Preis* (English “price” or “award”) from *Diesen Preis haben heute ... davongetragen*.

[1] `http://www.statmt.org/wmt16/translation-task.html`

State q: This *award* was received today by ...
State r: This *price* was received today by ...

Chain: {*Nobelpreis*, *Preis*, *Preisträger*}

Figure 2: Translation output of two different Docent states after applying the operation `change-phrase-translation`. Each state considers a different translation candidate of the German word *Preis*.

Since *Preis* is linked to *Nobelpreis* ("Nobel Prize") and *Preisträger* ("prize winner") in the source lexical chain, the semantic similarity of its counterpart lexical chain in the target is higher when *Preis* is translated into *award*. This leads to a higher LCTM score that contributes to a higher document score. The State q is then preferred by the decoder. Note that in this case, the language model also increases in State q. That is because *received* has a higher probability together with *award* than with *price*.

3.2.2 Computation of the Model Score

Each lexical chain is a chain of words connected by their semantic similarity, which is also computed using word embeddings. We define the model score as the mean of the semantic similarity scores of each target lexical chain in a document translation. To compute the semantic similarity *sim*$_i$ of a lexical chain *ch*$_i$, we average the semantic similarity of all links in *ch*$_i$ as in the following Equation

$$sim_i = \frac{1}{m} \sum_{j=1}^{m} SemLink_{ij}, \qquad (2)$$

where every link is comprised of two words, and its semantic similarity *SemLink* is the cosine similarity between their embeddings. In the experiments, we use German in the source, which is a language rich in compounds. These compounds have multiword equivalents in English and can be detected as part of a lexical chain (e.g. *Nordwand* is translated into the English *north face*). To deal with such cases, sim_i is the maximum similarity score obtained from each content word in the translation of a compound and the rest of the words in the lexical chain.

Every lexical chain has a different relevance in the computation of the LCTM score, which depends on three factors introduced by Morris and Hirst (1991): length (λ), repetition (β), and density (ρ). The later is defined as the ratio of words in the lexical chain to all words in the fragment of text that it covers. Accordingly, the longer, the denser the lexical chain is and the more repetition it has, the higher its weight is in the computation of the overall model score. These factors have not been addressed in the literature when dealing with lexical chains. Morris and Hirst (1991) define the strength of lexical chains, but they do not use it in their experiments.

To compute the length, density, and repetition of every lexical chain (i.e. λ_{ch_i}, ρ_{ch_i} and β_{ch_i}) we proceed as follows. Let *rel* be the total number of semantic relations in a lexical chain *ch*$_i$, *rep* the total number of repetitions, and *span* the number of words in the fragment of the document between the head and the tail of *ch*$_i$. ρ_{ch_i} and β_{ch_i} are then computed by the following two Equations

$$\rho_{ch_i} = \frac{rel}{span}, \qquad (3)$$

$$\beta_{ch_i} = \frac{rep}{span}. \qquad (4)$$

Finally, the length λ_{ch_i} is the ratio of *rel* to the number of relations of the longest lexical chain detected. The longest lexical chain gets therefore the highest length value (i.e. 1.0) among all lexical chains in the document.

After computing all factor values for each lexical chain, the model computes the weight for each of them. The weight of a chain w_{ch_i} is then the average of ρ_{ch_i}, λ_{ch_i} and β_{ch_i}, where ρ_{ch_i}, λ_{ch_i}, β_{ch_i}, and w_{ch_i} are all values between 0 and 1.[2]

Finally, the overall LCTM score is computed by

$$LCTM = \frac{1}{n} \sum_{i=1}^{n} w_{chi} \cdot \frac{1}{m_i} \sum_{j=1}^{m_i} SemLink_{ij}. \qquad (5)$$

3.3 Computation of Semantic Similarity

Dictionaries have been described in the literature to deal not only with lexical chains (Galley and McKeown, 2003), but with any task related to semantics such as WSD. However, it is unrealistic to assume that the fine-grained classification of

[2] We evaluated the impact of length, density, and repetition on translation by allowing tunable weights (0.0, 0.5, or 1.0) to each parameter and computing w_{ch_i} as the weighted average. The translation differences between the configurations were small, and the best performance was obtained when all of them had the maximum weight (1.0).

senses in dictionaries is adequate for any NLP application (Kilgarriff, 2006). Even the classification itself has been questioned in terms of cognitive validity (Kilgarriff, 1997, 2006; Hanks, 2000).

As Firth (1957) stated "You shall know a word by the company it keeps". That is, words that are used and occur in the same contexts tend to have similar meanings. Essentially, word embeddings are vector representations of words in a vector space that are learned based on the immediate context in which they occur. Our method uses word embeddings as a means to compute semantic similarity between words independently of dictionary senses to detect the lexical chains in the source and to compute the LCTM score.

The coverage and the quality of the lexical chains are the most important factors in our approach to improve translation. Words that are not in any lexical chain are not considered for improvement at the decoding stage by our LCTM. Word embeddings detect words as semantically related when they occur in similar context, even if they do not have a hypernym, hyponym or sibling relation. Halliday and Hasan (2014) define the words that do not have a traditional sense relation, but belong to the notion of lexical cohesion as *collocations*. The lexical chain detection method includes them in the same lexical chain, since they also help to disambiguate the translation of a word. For example, the word *climber* can be related to *mountain* with word embeddings, but not with Galley and McKeown (2003)'s approach.

The main problem of word embeddings arises from words with multiple senses that are not disambiguated in the training phase. That is, each word has only one vector representation, including those polysemic words. For example, consider the English word *play*, which appears in different contexts such as to perform on a musical instrument, to take part in a sport or game, and to interpret a role. The word embedding then represents all senses together. Consequently, the semantic similarity between *play* and *guitar* is low, because the similarity is computed between *guitar* and all the senses of *play* together.

Word senses need to be disambiguated in the training phase to generate distinct vector representations for each sense. We therefore employ a method introduced by Thater et al. (2011), which uses the syntactic information to build *contextualized* embeddings.[3] Consider again the word *play*, which appears in the sentences *we play the piano*, *we play the guitar*, *we play tennis*, *they play football*, and *they play Hamlet*. Following the approach proposed by Thater et al. (2011), we extract all the syntactic relations such as subject or object and group sentences in the same context by computing the semantic similarity between the context words (e.g. *piano* and *guitar*). As a result, we obtain (1) *we play the piano*, *we play the guitar*; (2) *we play tennis*, *they play football*; and (3) *they play Hamlet*. Lastly, we build the corresponding word embeddings *play_piano* for play the piano and the guitar, *play_tennis* for play tennis and football, and *play_Hamlet*.

	Training	Tuning	LM
Lines	400K	5K	570K
Tokens	∼ 11M	∼ 125K	∼ 15M

Table 1: Total of segments per language pair from Europarl and News Commentary used to train the German→English phrase-based SMT system.

Finally, to compute the semantic similarity of two words, our method computes the cosine similarity between their vector representation. The closer to 1.0 the resulting value is, the more similar they are. We set a threshold of 0.45 to distinguish between similar and non-similar words. This threshold is manually picked by looking at how different values impact on the resulting lexical chains. A lower threshold introduces too many words that are mostly related by their part-of-speech. A higher threshold results in semantically strong lexical chains, but it misses out on words that are also related.

4 Task Setup

We conducted several experiments to prove the efficacy of the lexical chain detection and LCTM in SMT. Lexical chains are difficult to evaluate in isolation, and therefore their quality is usually evaluated on the basis of the application for which they are used. Thus, we assess the performance of the method on the German→English translation task.

We then compare it to the algorithm presented by Galley and McKeown (2003), which uses external resources instead of word embeddings to build

[3] Any method that disambiguates the word senses and computes their word embeddings accordingly could be used.

the lexical chains. To build the lexical chains following Galley and McKeown (2003)'s method, we use GermaNet (Hamp and Feldweg, 1997) as external resource on the German side. The detected lexical chains are automatically annotated in the MMAX format[4] and then fed into Docent.

The data comes from the shared WMT'16 translation task.[5] We build a German→English phrase-based SMT system with Moses using standard settings (Koehn et al., 2003), 5-gram language model KenLM (Heafield, 2011) and GIZA++ (Och and Ney, 2003). The system is trained on Europarl, a parallel corpus of the proceedings of the European Parliament and News Commentary in equal parts (see Table 1). We use the first 17 documents of newstest2011 (554 segments), newstest2012 (684 segments), and newstest2013 (1,053 segments) for testing and newstest2010 (375 segments) as a development set of the LCTM and LCTM$_{base}$.

The method uses word embeddings to detect the source lexical chains. We therefore train a skip-gram 300-dimensional model in German using the $word2vec$ tool.[6] The texts come mainly from SdeWaC (Faaß and Eckart, 2013) ($\sim$768M words) and Common Crawls ($\sim$775M words). The rest of the data is from Europarl ($\sim$47M words) and News Commentary ($\sim$6M words). The LCTM model also needs to compute the similarity of the words in the target lexical chains. For this purpose, we employ a skip-gram 300-dimensional model trained on English Google News ($\sim$100 billion words).[6]

5 Experimental Results

In this section, we present the results obtained through the combination of lexical chain detection (using word embeddings and GermaNet) and the LCTM. The LCTM takes into account the relevance (i.e. strength) of every lexical chain to compute the overall score. We also perform a third experiment that ignores this fact to assess its impact in the translation quality. To do so, we develop a model that behaves like the LCTM, except that it assigns the maximum strength value (i.e. 1.0) to all lexical chains. We refer to this new model in the following as LCTM$_{base}$.

The baseline BLEU scores (Papineni et al., 2002) of the test sets newstest2010, newstest2011, and newstest2013 are 12.44, 12.18, and 17.64, respectively. The results of the experiments show between 20 to 30 translation changes in every test set due to lexical chains. We observe that the translation changes are often correct although they do not use the same terms as in the reference. Therefore, the fluctuations in BLEU scores are small (±0.1), and so BLEU does not provide sufficient insight into the performance.

Chain	***politik*** → *politischer*
Input	ich bin ein neuling in der prager **politik**
Ref.	i'm a novice in prague ***politics***
Base.	i am a newcomer in the prague ***policy***
LC	i am a newcomer in the prague ***politics***
Chain	*erklärt* → ***meint*** → *meint*
Input	"hier geht niemand vor gericht", **meint** . . .
Ref.	"nobody will sue them here," ***said***. . .
Base.	"here is no one in court", . . .
LC	"here is no one in court", ***says***. . .
Chain	*rakete* → ***rakete*** → *motor*
Input	. . . technische schäden an der **rakete**
Ref.	. . . technical damage to the ***missile***
Base.	. . . technical damage to the ***rocket***
LC	. . . technical damage to the ***missile***
Chain	*erhöht* → ***lohn*** → *lohnerhöhungen*
Input	. . . mehr als sie für **lohn** spenden.
Ref.	. . . more than it spends on ***salaries***.
Base.	. . . more than they for ***wage*** donations.
LC	. . . more than they for ***pay*** donations.

Figure 3: In these examples, the method produces a correct translation of the ambiguous word *Politik*, forces the translation of the German verb *meint*, and generates another good translation of *Rakete*. In the last example, the presented method incorrectly translates *lohn* into *pay*, despite the context given by the lexical chain: *ehöht* ("increase") and *lohnerhöhungen* ("wage increases").

We then perform a manual evaluation to assess the results of the experiments. The annotation is carried out by two annotators who judge the quality of the translation changes due to the lexical chains. Specifically, the annotators obtain for each translation change the source sentence, the baseline (i.e. the translation ignoring lexical chains), the translation produced by the method we want to evaluate, and the reference. They then anno-

[4] http://mmax2.sourceforge.net

[5] http://www.statmt.org/wmt16/translation-task.html

[6] https://code.google.com/p/word2vec

	newstest2011			newstest2012			newstest2013		
	+	−	++	+	−	++	+	−	++
Word Emb. & LCTM (1)	0.81	0.19	0.48	0.88	0.12	0.36	0.83	0.17	0.39
GermaNet & LCTM (2)	0.71	0.29	0.38	0.62	0.38	0.31	0.65	0.35	0.35
Word Emb. & LCTM$_{base}$ (3)	0.64	0.36	0.22	0.67	0.33	0.18	0.61	0.39	0.16

Table 2: Manual evaluation results of the presented method (1) compared to using GermaNet for lexical chain detection (2). The analysis shows the percentage of correct (+), wrong translations (-), and the improvement over the baseline (++). There are a total of 20 to 30 translation changes in every test set due to the lexical chains. We observe that the method (1) outperforms the approach that uses GermaNet (2). It also performs better than the method that ignores length, density, and repetition for the computation of the strength of each lexical chain in the overall score (3).

tate whether the word that changes due to lexical chains is better than the one produced by the baseline, equally good or worse. The Cohen's Kappa coefficient of inter-rater agreement between the two annotators is 0.77 (Cohen, 1960). We then compute from the annotations the percentage of incorrect and good translations and the improvement over the baseline.

Table 2 shows the results of the manual evaluation. We observe that the combination of lexical chain detection using word embeddings with our LCTM performs best. In particular, 81%-88% of the changes are correct translations, and among them, 36%-48% are improvements over the baseline. Only 12%-19% of the changes are incorrect. With GermaNet to detect lexical chains, the correctness decreases between 10% and 26%. Word embeddings may work better than lexical resources as they capture contextual information from the text, without relying on whether is defined in a resource. In those cases, where the resource does not provide a relation for two given words such as in idiomatic or metaphoric uses, the lexical chain cannot benefit from them.

The parameters length, density, and repetition have an impact on translation when using them to compute the strength of each lexical chain in the overall LCTM score. We see that the correctness of the translation output decreases approximately by 20% in all test sets when using the LCTM$_{base}$ (i.e. the model that gives the highest strength value to all lexical chains, ignoring the mentioned parameters) instead of the LCTM. Furthermore, the percentage of the improvements over the baseline decrease by half.

Some translation examples using our method are illustrated in Figure 3. In the first example, the ambiguous German noun *Politik* gets correctly translated into *politics*. *Politik* is connected to *politischer* ("political") in the lexical chain, and therefore *politics* is semantically more related to *political* than *policy*. Our method is also good at enforcing the translation of all words in the lexical chain, since an untranslated word will decrease the score of the translated lexical chain, and accordingly, the overall LCTM score (see Example 2). In the last example, the method produces a wrong translation of the German word *lohn* ("wage", "salary"), whereas the baseline translates it correctly. The word *lohn* is linked to *erhöht* ("increase") and *lohnerhöhungen* ("wage increases") in the lexical chain. Both words provide good context for the translation. However, our method incorrectly translates it into *pay*, whereas the baseline translates it correctly into *wage*.

In the third example, we observe that the method produces a different but equally good translation compared to the baseline. In the lexical chain, the German word *Rakete* is linked to another occurrence of the same word that is translated into *missile*. Since the highest similarity score is obtained when both translations are the same, our method encourages consistency, translating both into *missile* (Carpuat, 2009; Carpuat and Simard, 2012). Consistency is possible since we assume that there is only a unique sense per word in each document (Gale et al., 1992).

Figure 4 illustrates the benefits and issues of consistent translation. These are special cases, where the word in the lexical chain is linked only to other occurrences of the same word.

In the first example, we observe that the baseline translates the wrong sense of the word *wahl* (i.e. *choice*). Here, *wahl* is linked to another oc-

Input	er entschloss . . . , sich an der **wahl** vor der letzten hauptversammlung zu beteiligen
Ref.	he decided to participate in the ***elections*** before the last general meeting . . .
Base.	he decided . . . , the ***choice*** of the last hauptversammlung to participate
LC	he decided . . . , the ***election*** of the last hauptversammlung to participate
	Linked to:
Input	. . . für die heutigen probleme mit der **wahl** die euphorie verantwortlich ist . . .
Ref.	. . . current problems with ***elections*** are caused by the euphoria there was . . .
LC	. . . for today's problems, with the ***election*** of the euphoria is responsible . . .

Input	das **verhältnis** der länge der beiden erwähnten finger . . .
Ref.	the ***ratio*** of the length of those two fingers . . .
Base.	the ***ratio*** of the length of the two . . .
LC	the ***relationship*** between the length of the two . . .
	Linked to:
Input	. . . dennoch halte er das **verhältnis** zwischen der fingerlänge und dem krebsrisiko . . .
Ref.	. . . but in his opinion the ***relationship*** between the length of the fingers and the cancer . . .
LC	. . . but it the ***relationship*** between the fingerlänge and the risk of cancer . . .

Figure 4: These examples show how the presented method behaves when a word in the lexical chain is linked to the same word in the text. In the first example, the German word *wahl* is linked to another occurrence of *wahl* in the text. The later is correctly translated into *election*, and therefore the LCTM gets a higher score when the first sentence is translated into the same term. This produces an improvement over the baseline that wrongly translates it into *choice*. In the second example, both senses of the word *verhältnis* occur in the same document, forcing the first occurrence to be incorrectly translated.

currence of the same word in the lexical chain, which is translated into the other sense *election*. Since the method obtains the highest score when the translations are the same, it either encourages both occurrences to be translated into *election* or *choice*. The LCTM score competes with other models such as language and translation model. The overall score when using the translation *choice* is then lower than when using *election* due to the other models, since *choice* does not fit in the local context of the other sentence.

In the second example, however, the method translates the wrong sense of *verhältnis*. That is because the two senses of the word *verhältnis* ("ratio" and "relationship") are in the same document. This fact violates the one-sense-per-discourse hypothesis, and when the only context provided by the lexical chain is the word itself, the method cannot disambiguate the senses.

6 Summary and Conclusions

We present a method that utilises lexical chains to improve the quality of document-level SMT output, showing that the translation improves when discourse knowledge is considered. Specifically, the method improves the translation of the words in the chains, keeping the semantic similarity from the source to the translation. Each lexical chain captures a portion of the cohesive structure of a document. It is therefore essential to ensure that the words in the lexical chains are well translated.

The method is divided into two steps that consist of detecting the lexical chains in the source and preserving the semantic similarity among the words in their counterpart target lexical chains. We therefore implement an automatic detection of the lexical chains based on a manual approach proposed by Morris and Hirst (1991) and a feature function in the document-level decoder Docent (i.e the LCTM) that preserves the semantic similarity in the translated chains.

Our method uses word embeddings instead of external lexical resources to deal with word similarity. To detect the similarity between polysemic words, we need to disambiguate words in the training phase. We therefore apply the approach described by Thater et al. (2011), which relies on syntactic information to differentiate a word that appears in different contexts.

We assess the performance of the lexical chain detection on the translation task. The manual evaluation of the results show that the proposed method improves between 36% and 48% of the changes over a baseline that does consider lexical chains or any document-level knowledge. The results of the method are also evaluated against the method proposed by Galley and McKeown (2003), which uses a dictionary instead.

The method shows a bias for consistently translating the words in the chain. Since we assume the one-sense-per-discourse hypothesis (Gale et al., 1992), this is the preferred behaviour. Here, the method has the advantage that during decoding the LCTM competes with other feature functions. Therefore, the decoder favours the consistent translation of the repeated words in a chain that fits in all their contexts, avoiding consistently translating the wrong sense.

When the one-sense-per-discourse hypothesis does not hold, different senses of the same word may be linked in the same lexical chain. This poses a problem when each sense has a different translation in the target language. The method cannot distinguish between different senses and incorrectly translates them in the same way.

The lexical chains detected in the source differ from each other in length, density, and total of repetitions. To ensure that they have a different degree of impact on translation depending on their strength in the document, the LCTM takes that into account in the computation of the model score. To assess the importance of distinguishing between lexical chains, we implement a simplified version of the LCTM (LCTM$_{base}$) that gives the same strength value to all chains in the document. The experimental results show that the method that uses the LCTM$_{base}$ performs worse that LCTM in all test sets.

Acknowledgments

The author would like to thank Don Tuggener and Annette Rios for the helpful discussions and the manual annotation. This research was supported by the Swiss National Science Foundation under the Sinergia MODERN project (grant n. 147653, see www.idiap.ch/project/modern/).

References

Regina Barzilay and Michael Elhadad. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain, pages 10–17.

Marine Carpuat. 2009. One Translation Per Discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Boulder, CO, USA, pages 19–27.

Marine Carpuat and Michel Simard. 2012. The Trouble with SMT Consistency. In *Proceedings of the 7th Workshop on Statistical Machine Translation*. Montréal, Canada, pages 442–449.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A Unified Model for Word Sense Representation and Disambiguation. In *Proceedings of the 13th Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar, pages 1025–1035.

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA, pages 263–270.

Jacob Cohen. 1960. A coefficient of agreement for nominal scale. *Educational and Psychological Measurement* 20:37–46.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – A Corpus of Parsable Sentences from the Web. In *Language Processing and Knowledge in the Web*, Springer Berlin Heidelberg, volume 8105, pages 61–68.

John R Firth. 1957. A Synopsis of Linguistic Theory, 1930-1955. Blackwell.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. One Sense per Discourse. In *Proceedings of the Workshop on Speech and Natural Language*. Harriman, NY, USA, pages 233–237.

Michel Galley and Kathleen McKeown. 2003. Improving Word Sense Disambiguation in Lexical Chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA, pages 1486–1488.

Eva Martínez Garcia, Carles Creus, Cristina España i Bonet, and Lluís Màrquez. 2017. Using Word Embeddings to Enforce Document-Level Lexical Consistency in Machine Translation. DE GRUYTER OPEN, volume 108, pages 85–96.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based Document-level Statistical Machine Translation. In *Proceedings of the 10th Conference on Empirical Methods in Natural Language Processing*. Edinburgh, UK., pages 909–919.

Liane Guillou. 2013. Analysing Lexical Consistency in Translation. In *Proceedings of the Workshop on Discourse in Machine Translation*. Sofia, Bulgaria, pages 10–18.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in English*. Routledge.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of the Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. Madrid, Spain, pages 9–15.

Patrick Hanks. 2000. Do word meanings exist? *Computers and the Humanities* 34(1):205–215.

Christian Hardmeier, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. 2013. Docent: A Document-Level Decoder for Phrase-Based Statistical Machine Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 193–198.

Eva Hasler, Barry Haddow, and Philipp Koehn. 2014. Dynamic Topic Adaptation for SMT using Distributional Profiles. In *Proceedings of the 9th Workshop on Statistical Machine Translation*. Baltimore, MD, USA, pages 445–456.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*. Edinburgh, Scotland, pages 187–197.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, pages 897–907.

Adam Kilgarriff. 1997. I Don't Believe in Word Senses. *Computers and the Humanities* 31:91–113.

Adam Kilgarriff. 2006. *Word Senses*. Springer Netherlands, Dordrecht.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*. pages 177–180.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of the 4th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Edmonton, Canada, pages 48–54.

Laura Mascarell, Mark Fishel, Natalia Korchagina, and Martin Volk. 2014. Enforcing Consistent Translation of German Compound Coreferences. In *Proceedings of the 12th Konvens Conference*. Hildesheim, Germany, pages 58–65.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the Workshop at the International Conference on Learning Representations*. Scottsdale, AZ, USA.

George A Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38(11):39–41.

Jane Morris and Graeme Hirst. 1991. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics* 17(1):21–48.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Sapporo, Japan.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, USA, pages 311–318.

Mohsen Pourvali and Mohammad Saniee Abadeh. 2012. Automated Text Summarization Base on Lexicales Chain and Graph using of WordNet and Wikipedia Knowledge Base (sic!). *Computing Research Repository* .

Xiao Pu, Laura Mascarell, and Andrei Popescu-Belis. 2017. Consistent Translation of Repeated Nouns using Syntactic and Semantic Cues. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain, pages 948–957.

Xiao Pu, Laura Mascarell, Andrei Popescu-Belis, Mark Fishel, Ngoc-Quang Luong, and Martin Volk. 2015. Leveraging Compounds to Improve Noun Phrase Translation from Chinese and German. In *Proceedings of the ACL-IJCNLP Student Research Workshop*. Beijing, China, pages 8–15.

Antonio M. Rinaldi. 2009. An Ontology-Driven Approach for Semantic Information Retrieval on the Web. *ACM Transactions on Internet Technology* 9(3):10.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China, pages 1793–1803.

Mark Stairmand. 1996. *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval*. Ph.D. thesis, University of Manchester.

Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. In *Proceedings of the 14th Conference of the North American Chapter of the Association for Computational Linguistics*. Denver, Colorado, pages 314–323.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand, pages 1134–1143.

Jörg Tiedemann. 2010. Context Adaptation in Statistical Machine Translation Using Models with Exponentially Decaying Cache. In *Proceedings of the Workshop on Domain Adaptation for Natural Language Processing*. Uppsala, Sweden, pages 8–15.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level Consistency Verification in Machine Translation. In *Proceedings of the 13th Machine Translation Summit*. Xiamen, China, pages 131–138.

Deyi Xiong, Yang Ding, Min Zhang, and Chew Lim Tan. 2013. Lexical Chain Based Cohesion Models for Document-Level Statistical Machine Translation. In *Proceedings of the 12th Conference on Empirical Methods in Natural Language Processing*. Seattle, WA, USA, pages 1563–1573.

Deyi Xiong and Min Zhang. 2013. A Topic-based Coherence Model for Statistical Machine Translation. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*. Bellevue, WA, USA.

Deyi Xiong and Min Zhang. 2014. A Sense-Based Translation Model for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, MD, USA, pages 1459–1469.

Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. System Demonstrations*. Uppsala, Sweden, pages 78–83.