Lexical Chain Based Cohesion Models for Document-Level Statistical Machine Translation

Deyi Xiong[1], **Yang Ding**[2], **Min Zhang**[1*] **and Chew Lim Tan**[2]
[1]School of Computer Science and Technology, Soochow University, Suzhou, China 215006
{dyxiong, minzhang}@suda.edu.cn
[2]School of Computing, National University of Singapore, Singapore 117417
{a0082379, tancl}@comp.nus.edu.sg

Abstract

Lexical chains provide a representation of the lexical cohesion structure of a text. In this paper, we propose two lexical chain based cohesion models to incorporate lexical cohesion into document-level statistical machine translation: 1) a count cohesion model that rewards a hypothesis whenever a chain word occurs in the hypothesis, 2) and a probability cohesion model that further takes chain word translation probabilities into account. We compute lexical chains for each source document to be translated and generate target lexical chains based on the computed source chains via maximum entropy classifiers. We then use the generated target chains to provide constraints for word selection in document-level machine translation through the two proposed lexical chain based cohesion models. We verify the effectiveness of the two models using a hierarchical phrase-based translation system. Experiments on large-scale training data show that they can substantially improve translation quality in terms of BLEU and that the probability cohesion model outperforms previous models based on lexical cohesion devices.

1 Introduction

Given a source document, traditionally most statistical machine translation (SMT) systems translate the document sentence by sentence. In such a translation scheme, sentences are translated independent of any other sentences. However, a text is normally written cohesively, in which sentences are connected to each other via syntactic and lexical devices. This linguistic phenomenon is called as *textual cohesion* (Halliday and Hasan, 1976).

Cohesion is a surface-level property of well-formed texts. It deals with five categories of relationships between text units, namely co-reference, ellipsis, substitution, conjunction and lexical cohesion that is realized via semantically related words. The former four cohesion relations can be grouped as grammatical cohesion. Generally speaking, grammatical cohesion is less common and harder to identify than lexical cohesion (Barzilay and Elhadad, 1997).

As most SMT systems translate a text in a sentence-by-sentence fashion, they tend to build less lexical cohesion than human translators (Wong and Kit, 2012). We therefore study lexical cohesion for document-level translation. We use lexical chains (Morris and Hirst, 1991) to capture lexical cohesion in a text. Lexical chains are connected graphs that represent the lexical cohesion structure of a text. They have been successfully used for information retrieval (Stairmand, 1996), document summarization (Barzilay and Elhadad, 1997) and so on. In this paper, we investigate how lexical chains can be used to incorporate lexical cohesion into document-level translation.

Our basic assumption is that the lexical chains of a target document are direct correspondences of the lexical chains of its counterpart source document. This assumption is reasonable as the target document translation should be faithful to the source document in terms of both text meaning and structure. Based on this assumption, we propose a framework

[*] Corresponding author

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1563–1573,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

to incorporate lexical cohesion into target document translation via lexical chains, which works as follows.

- Compute lexical chains for each source document that is to be translated;
- Project the computed source lexical chains onto the corresponding target document by translating source chain words into target chain words using maximum entropy classifiers;
- Incorporate lexical cohesion into the target document translation via cohesion models built on the projected target lexical chains .

We build two lexical chain based cohesion models. The first model is a count model that rewards a hypothesis whenever a word in the projected target lexical chains occur in the hypothesis. As a source chain word may be translated into many different target words, we further extend the count model to a second cohesion model: a probability model that takes chain word translation probabilities into account.

We test the two lexical chain based cohesion models on a hierarchical phrase-based SMT system that is trained with large-scale Chinese-English bilingual data. Experiment results show that our lexical chain based cohesion models can achieve substantial improvements over the baseline. Furthermore, the probability cohesion model is better than the count model and it also outperforms previous cohesion models based on lexical cohesion devices (Xiong et al., 2013).

To the best of our knowledge, this is the first attempt to explore lexical chains for statistical machine translation. The remainder of this paper is organized as follows. Section 2 discusses related work and highlights the differences between our method and previous work. Section 3 briefly introduces lexical chains and algorithms that compute lexical chains. Section 4 elaborates the proposed lexical chain based framework, including details on source lexical chain computation, target lexical chain generation and the two lexical chain based cohesion models. Section 5 presents our large-scale experiments and results. Finally, we conclude with future directions in Section 6.

2 Related Work

Recent years have witnessed growing research interests in document-level statistical machine translation. Such research efforts can be roughly divided into two groups: 1) general document-level machine translation that does not explore or explores very little linguistic discourse information; 2) linguistically-motivated document-level machine translation that incorporates discourse information such as cohesion and coherence into SMT. Recent studies (Guillou, 2013; Beigman Klebanov and Flor, 2013) show that this discourse information is very important for document-level machine translation.

General Document-Level Machine Translation Tiedemann (2010) propose cache-based language and translation models for document-level machine translation. These models are built on recently translated sentences. Following this cache-based approach, Gong et al. (2011) further introduce two additional caches. They use a static cache to store bilingual phrases extracted from documents in training data that are similar to the document being translated. They also adopt a topic cache with target language topic words. Xiao et al. (2011) study the translation consistency issue in document-level machine translation. They use a hard constraint to consistently translate ambiguous source words into the most frequent translation options. Ture et al. (2012) soften this consistency constraint by integrating three counting features into decoder.

Using Lexical Cohesion Devices in Document-Level SMT Lexical cohesion devices are semantically related words, including word repetition, synonyms/near-synonyms, hyponyms and so on. They are also the cohesion-building elements in lexical chains.

Wong and Kit (2012) use lexical cohesion device based metrics to improve machine translation evaluation at the document level. These metrics measure the proportion of content words that are used as lexical cohesion devices in machine-generated translations. Hardmeier et al. (2012) propose a document-wide phrase-based decoder and integrate a semantic language model into the decoder. They argue that their semantic language model can capture lexical cohesion by exploring n-grams that cross sentence boundaries.

Most recently Xiong et al. (2013) integrate three categories of lexical cohesion devices into document-level machine translation. They define three cohesion models based on lexical cohesion devices: a direct reward model, a conditional probability model and a mutual information trigger model. The latter two models measure the strength of lexical cohesion relation between two lexical items. They are incorporated into SMT to calculate how appropriately lexical cohesion devices are used in document translation. As lexical chains capture lexical cohesion relations among sequences of related words rather than those only between two words, experiments in Section 5 show that our lexical chain based probability cohesion model is better than the lexical cohesion device based trigger model, which is the best among the three cohesion models proposed by Xiong et al. (2013).

Modeling Coherence in Document-Level SMT In discourse analysis, cohesion is often studied together with *coherence* which is another dimension of the linguistic structure of a text (Barzilay and Elhadad, 1997). Cohesion is related to the surface structure of a text while coherence is concerned with the underlying meaning connectedness in a text (Vasconcellos, 1989). Compared with cohesion, coherence is not easy to be detected. Even so, various models have been proposed to explore coherence for document summarization and generation (Barzilay and Lapata, 2008; Louis and Nenkova, 2012). Following this line, Xiong and Zhang (2013) integrate a topic-based coherence model into document-level machine translation, where coherence is defined as a continuous sentence topic transition.

Our lexical chain based cohesion models are also related to previous work on using word and phrase sense disambiguation for lexical choice in SMT (Carpuat and Wu, 2007b; Carpuat and Wu, 2007a; Chan et al., 2007). The difference is that we use document-wide lexical chains to build our cohesion models rather than sentence-level context features. In our framework, lexical choice is performed to make the selected words consistent with the lexical cohesion structure of a document.

Carpuat (2009) explores the principle of one sense per discourse (Gale et al., 1992) in the context of SMT and imposes the constraint of one translation per discourse on document translation. We also use the one sense per discourse principle to perform word sense disambiguation on the source side in our lexical chaining algorithm (See Section 4.1).

3 Background: Lexical Chain and Chain Computation

Lexical chains are sequences of semantically related words (Morris and Hirst, 1991). They represent the lexical cohesion structure of a text. Figure 2 displays six lexical chains computed from the Chinese news article shown in Figure 1. Words in these lexical chains have lexical cohesion relations such as repetition, synonym, which may range over the entire text. For example, in the lexical chain $\textsc{lc}^1$ of Figure 2, the same word "déguó" (Germany) repeats 9 times. In the lexical chain $\textsc{lc}^3$, the two words "zǒngcǎi" (president) and "zhǔxí" (chairman) are synonym words. Generally, a text can have many different lexical chains, each of which represents a thread of cohesion through the text.

Several lexical chaining algorithms have been proposed to compute lexical chains from texts. Normally they need an ontology to obtain semantic relations between words. Word sense disambiguation (WSD) is also used to determine the sense of each word in a text. Generally a lexical chain computation algorithm completes the following three subtasks:

- Building a representation of a text with a set of candidate words and assigning semantic relations between the candidate words according to the ontology;
- Choosing the right sense for each candidate word via WSD;
- Building chains over the semantically related and disambiguated candidate words.

These three sub-tasks can be done separately or simultaneously.

Morris and Hirst (Morris and Hirst, 1991) define the first lexical chain computation algorithm that adopts a greedy strategy to immediately disambiguate a word at its first occurrence. This algorithm runs in linear time but suffers from inaccurate disambiguation. Barzilay and Elhadad (Barzilay and Elhadad, 1997) significantly improve WSD

déguó diànxìn gōngsī zŏngcái suŏmò cízhí
déguó diànxìn gōngsī xuānbù , qián jiānshìhuì zhŭxí qīshíèr suì de xīlèĕr jiāng dānrèn gāi gōngsī de línshí zŏngcái , wéiqī liù gè yuè , zhídào suŏmò de jìrèn rénxuăn jiērèn wéizhĭ 。
(făxīnshè bō áng diàn) déguó diànxìn gōngsī zŏngcái suŏmò jīntiān cíqù tā de zhíwù , tā shuō , yóuyú tā xiănrán bú zài shòudào déguó diànxìn gōngsī jiānshìhuì de chōngfèn xìnrèn , cízhí shì tā wéiyī de xuănzé 。
tóuzīrén huānyíng zhèxiàng xuānbù , déguó diànxìn gōngsī de gŭpiào yīncĭ zài fălánkèfú gŭpiào jiāoyì shìchăng shàng zhăng băifènzhīshíyī yĭshàng 。
suŏmò zài déguó diànxìn gōngsī bō áng zŏngbù zhàokāi jiānshìhuì tèbié huìyì zhōng fābiăo yī xiàng shēngmíng , tā shuō : 「 wŏ yĭ yàoqiú jiānshìhuì jiĕchú wŏ de zhíwù 。 」
yóuyú liăng gè yuè hòu jiāng jŭxíng dàxuăn , dàn liánhézhèngfŭ zài mín diào zhōng shēngwàng luòhòu , déguó zŏnglĭ shī ruòdé sìhū xīwàng zài gŭjià xiàcuò zhì xīn dī shí , déguó diànxìn gōngsī shùbăiwàn míng xiăo gŭdōng de zījīn bìng wèi xiāoshī , ér zhīchí tā 。
déguó diànxìn gŭjià hòulái huí wĕn , yĭ shíyīdiănyībā ōuyuán zuò shōu , shàngzhăng băifènzhībādiănwŭsì 。
déguó cáizhèngbù huānyíng suŏmò cízhí de juédìng 。

Figure 1: An example of a Chinese news article (written in pinyin).

LC^1: {déguó, déguó, bō, déguó, déguó, déguó, déguó, bō, déguó, déguó, déguó}
LC^2:{jiānshìhuì, făxīnshè, jiānshìhuì, zŏngbù, jiānshìhuì, jiānshìhuì}
LC^3: {zŏngcái, zhŭxí, zŏngcái, zŏnglĭ}
LC^4: {cízhí, cíqù, cízhí, cízhí}
LC^5: {zhăng, xiàcuò, shàngzhăng}
LC^6: {xuānbù, xuānbù, fābiăo}

Figure 2: Six lexical chains from the example in Figure 1.

accuracy by processing all possible combinations of word senses in a text to disambiguate words. Unfortunately, their algorithm runs slowly in quadratic time. Galley and Mckeown (2003) present an algorithm that are better than the former two algorithms both in terms of running efficiency and WSD accuracy. They separate the WSD sub-task from the task of lexical chain building and impose a "one sense per discourse" constraint in the WSD step.

4 Translating Documents Using Lexical Chains

In this section, we describe how we incorporate lexical cohesion into document-level machine translation using lexical chains. We divide the lexical chain based document-level machine translation process into three steps: (1) computing lexical chains for source documents with a source language ontology, (2) generating target lexical chains from the computed source lexical chains, and finally (3) incorporating lexical cohesion encoded in the generated target lexical chains into document-level translation via lexical chain based cohesion models. The remainder of this section will elaborate these three steps.

4.1 Source Lexical Chains Computation

We follow the chain computation algorithm introduced by Galley and McKeown (2003) to build lexical chains on source (Chinese) documents. In the algorithm, the chaining process includes three steps: choosing candidate words to build a disambiguation graph (Galley and McKeown, 2003) for each document, disambiguating the candidate words and finally building lexical chains over the disambiguated candidate words.

The disambiguation graph can be considered as a representation of all possible interpretations of its corresponding text. In the graph, nodes are candidate words with different senses and edges between word senses are weighted according to their semantic relations, such as synonym, hypernym and so on. We use an extended version of a Chinese thesaurus *Tongyici Cilin* (Cilin for short) to define word senses and semantic relations between senses. The ex-

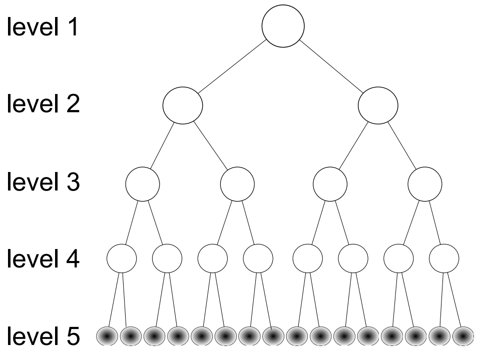

Figure 3: The architecture of the extended Cilin. For simplicity, we only draw a binary tree to represent the hierarchical structure of Cilin. This doesn't mean that each semantic class at level i has only two sub-classes at level $i+1$. Actually, they have multiple sub-classes.

tended Cilin contains 77,343 Chinese words, which are organized in a hierarchical structure containing 5 levels as shown in Figure 3. In the 5th level, each node represents an atomic concept which consists of a set of synonyms. These atomic concepts are just like *synsets* in WordNet. We use them to represent senses of words in the disambiguation graph.

We select nouns, verbs, abbreviations and idioms as candidate words for the disambiguation graph. These words are identified by a Chinese part-to-speech tagger LTP (Che et al., 2010) in a preprocessing step. In order to build the disambiguation graph, we first build an array indexed by the atomic concepts of Cilin, then insert a copy of each candidate word into its all concept (sense) entries in the array. After that, we create all semantic links among senses of different candidate words in the disambiguation graph following Galley and McKeown (2003).

In the second step, we use the principle of one sense per discourse to perform WSD for each candidate word in the disambiguation graph. We sum the weights of all semantic links under the different senses of the candidate word in question. The sense with the highest sum of weights is considered as the most probable sense for this word. We then assign this sense to all occurrences of the word in the document by adopting the constraint of one sense per discourse.

Once all candidate words are disambiguated, we can build lexical chains over these words by removing all semantic links that connect those unselected word senses. The six lexical chains shown in Figure 2 are computed from the Chinese document in Figure 1 exactly following the algorithm of Galley and McKeown (2003). The only difference is that we use Cilin rather than WordNet as the ontology.

4.2 Target Lexical Chains Generation

Since a faithful target document translation should follow the same cohesion structure as that in its corresponding source document, we generate target lexical chains from the computed source lexical chains. Given a source lexical chain $LC_s = \{s_i^j\}$ where the ith chain word s_i^j is from the jth sentence of the source document D_s, we generate a target lexical chain $LC_t = \{t_i^j\}$ using maximum entropy (MaxEnt) classifiers. Particularly, we translate a word s_i^j in the source lexical chain into a target word t_i^j in the target lexical chain using a corresponding MaxEnt classifier as follows[1].

$$P(t_i^j|\mathcal{C}(s_i^j)) = \frac{exp(\sum_k \theta_k f_k(t_i^j, \mathcal{C}(s_i^j)))}{\sum_t exp(\sum_k \theta_k f_k(t, \mathcal{C}(s_i^j)))} \quad (1)$$

where f_k are binary features, θ_k are weights of these features, and $\mathcal{C}(s_i^j)$ is the surrounding context of chain word s_i^j.

We train one MaxEnt classifier per unique source chain word. For each classifier, we define two groups of binary features: 1) the preceding and succeeding two words of s_i^j in the jth sentence ($\{w_{-2}, w_{-1}, s_i^j, w_{+1}, w_{+2}\}$); 2) the preceding and succeeding one word of s_i^j in the lexical chain LC_s ($\{s_{i-1}^p, s_i^j, s_{i+1}^q\}$). All features are in the following binary form.

$$f(t_i^j, \mathcal{C}(s_i^j)) = \begin{cases} 1, & \text{if } t_i^j = \clubsuit \text{ and } \mathcal{C}(s_i^j).\heartsuit = \spadesuit \\ 0, & \text{else} \end{cases} \quad (2)$$

where the symbol ♣ is a placeholder for a possible target word, the symbol ♡ indicates a contextual element for the chain word s_i^j (e.g., the preceding word in the jth sentence or the succeeding word in the lexical chain LC_s), and the symbol ♠ represents the value of ♡.

Given a source document D_s and its N lexical chains $\{LC_s^k\}_{k=1}^N$ computed from the document as

[1] We collect training instances from word-aligned bilingual data to train the MaxEnt classifier.

described in Section 4.1, we can generate the N target lexical chains $\{LC_t^k\}_{k=1}^N$ using our MaxEnt classifiers. Each target word t_i^j in the target lexical chain LC_t^k is the translation of its corresponding source word s_i^j in the source lexical chain LC_s^k with the highest probability $P(t_i^j|\mathcal{C}(s_i^j))$ according to Eq. (1).

As we know, the MaxEnt classifier can generate multiple translations for each source word. In order to incorporate these multiple chain word translations, we can generate a *super target lexical chain* ${}_\epsilon LC_t$ from a source lexical chain LC_s, where ϵ is a pre-defined threshold used to select multiple translations. For example, given a source lexical chain $LC_s = \{a, b, c\}$, we can have the corresponding super target lexical chain ${}_\epsilon LC_t = \{\{a_t^1, a_t^2 ...\}, \{b_t^1, b_t^2 ...\}, \{c_t^1, c_t^2 ...\}\}$, where x_t^i is the translation of x with a translation probability $P(x_t^i|\mathcal{C}(x)) \geq \epsilon$ according to Eq. (1). Integrating multiple translations for each source chain word, we can reduce the error propagation of the MaxEnt classifier to some extent. Our experiments also confirm that the super target lexical chains with multiple translation options for each chain word are better than the target lexical chains with only one translation per chain word. Therefore we build our cohesion models based on the super target lexical chains, which will be described in the next section.

4.3 Lexical Chain Based Cohesion Models

Once we generate the super target lexical chains $\{{}_\epsilon LC_t^k\}_{k=1}^N$ for the target document D_t, we can use them to provide constraints for the target document translation. Our key interest is to make the target document translation T_{D_t} as cohesive as possible. We therefore propose lexical chain based cohesion models to measure the cohesion of the target document translation. The basic idea is to reward a translation hypothesis if a word from the super target lexical chains occurs in the hypothesis. According to the difference in the reward strategy, we have two cohesion models: a *count cohesion model* and a *probability cohesion model*.

Count Cohesion Model $M_c(T_{D_t}, \{{}_\epsilon LC_t^k\}_{k=1}^N)$: This model rewards a translation hypothesis of the jth sentence in the document whenever a lexical chain word t_i^j occurs in the hypothesis. The model maintains a counter and accumulates the counter when necessary. It is factorized into the sentence cohesion metric $M_c(T_j, \{{}_\epsilon LC_t^k\}_{k=1}^N)$, where T_j is the translation of the jth sentence in the target document. $M_c(T_j, \{{}_\epsilon LC_t^k\}_{k=1}^N)$ is formulated as follows.

$$M_c(T_j, \{{}_\epsilon LC_t^k\}_{k=1}^N) = \prod_{w \in T_j} \prod_{t_i^j \in \mathbf{C}} e^{\delta(w, t_i^j)} \quad (3)$$

where $\mathbf{C}$ represents $\{{}_\epsilon LC_t^k\}_{k=1}^N$, and the δ function is defined as follows.

$$\delta(w, t_i^j) = \begin{cases} 1, & \text{if } t_i^j = w \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Probability Cohesion Model $M_p(T_{D_t}, \{{}_\epsilon LC_t^k\}_{k=1}^N)$: This model rewards a translation hypothesis according to the translation probability of a chain word that occurs in the hypothesis. The translation probability is computed by Eq. (1). The model is also factorized into the sentence cohesion metric $M_p(T_j, \{{}_\epsilon LC_t^k\}_{k=1}^N)$ which is formulated as follows.

$$M_p(T_j, \{{}_\epsilon LC_t^k\}_{k=1}^M) = \prod_{w \in T_j} \prod_{t_i^j \in \mathbf{C}} e^{\delta(w, t_i^j)} \times P(t_i^j|\mathcal{C}(s_i^j)) \quad (5)$$

where $P(t_i^j|\mathcal{C}(s_i^j)$ is the translation probability computed according to Eq. (1).

4.4 Decoding

The proposed lexical chain based cohesion models are integrated into the log-linear translation framework of SMT as a cohesion feature. Before translating a source document, we compute lexical chains for the source document as described in Section 4.1. We then generate the super target lexical chains. In order to efficiently calculate our lexical chain based cohesion models, we reorganize words in the super target lexical chains into vectors. We associate each source sentence S_j a vector to store target lexical chain words that are to occur in the corresponding target sentence T_j.

Although we still translate a source document sentence by sentence, we capture the global cohesion structure of the document via lexical chains and use the lexical chain based cohesion models to constrain word selection in document translation. Figure 4 shows the architecture of an SMT system with the lexical chain based cohesion model.

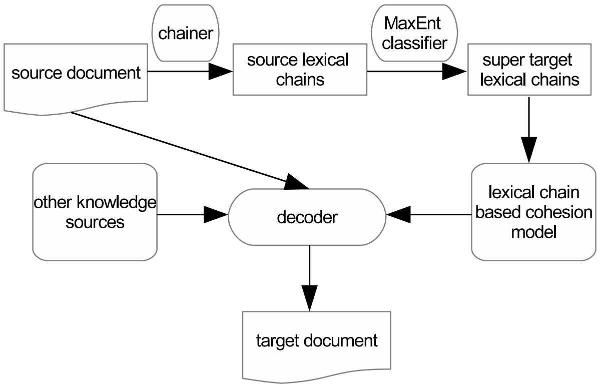

Figure 4: Architecture of an SMT system with the lexical chain based cohesion model.

5 Experiments

In this section, we conducted a series of experiments to validate the effectiveness of the proposed lexical chain based cohesion models for Chinese-to-English document-level machine translation. We used a hierarchical phrased-based SMT system (Chiang, 2007) trained on large-scale data. In particular, we aim at:

- Measuring the impact of the threshold ϵ on the probability cohesion model and selecting the best threshold on a development test set.
- Investigating the effect of the two lexical-chain based cohesion models.
- Comparing our lexical chain based cohesion models against the previous lexical cohesion device based models (Xiong et al., 2013).

5.1 Setup

We collected our bilingual training data from LDC, which includes the corpus LDC2002E18, LDC2003E07, LDC2003E14, LDC2004E12, LDC2004T07, LDC2004T08 (Only Hong Kong News), LDC2005T06 and LDC2005T10. The collected bilingual training data contains 3.8M sentence pairs with 96.9M Chinese words and 109.5M English words. We trained a 4-gram language model on the Xinhua portion of the English Gigaword corpus (306 million words) via the SRILM toolkit (Stolcke, 2002) with Kneser-Ney smoothing.

	Training	**MT05**	**MT06**	**MT08**
#Doc	103,236	100	79	109
#Sent	2.80M	1,082	1,664	1,357
#Chain	3.52M	1700	2172	1693
#AvgC	35.72	17	27.49	15.53
#AvgW	14.81	5.89	6.89	5.63

Table 1: Statistics of the training, development and test sets, which show the number of documents (#Doc) and sentences (#Sent), the number of lexical chains extracted from the source documents (#Chain), the average number of lexical chains per document (#AvgC) and the average number of words per lexical chain (#AvgW).

In order to build the lexical chain based cohesion models, we selected corpora with document boundaries explicitly provided from the bilingual training data together with the whole Hong Kong parallel text corpus as the cohesion model training data[2]. We show the statistics of these selected corpora in Table 1. They contain 103,236 documents and 2.80M sentences. Averagely, each document consists of 28.4 sentences. From the source documents of the selected corpora, we extract 3.52M lexical chains. On average, there are 35.72 lexical chains per document and 14.81 words per lexical chain.

We used the off-the-shelf MaxEnt toolkit[3] to train one MaxEnt classifier per unique source lexical chain word (61,121 different source chain words in total). We performed 100 iterations of the L-BFGS algorithm implemented in the training toolkit for each chain word with both Gaussian prior and event cutoff set to 1 to avoid overfitting. After event cutoff, we have an average of 17.75 different classes (target translations) per source chain word.

We used the NIST MT05 as the tuning set for the minimum error rate training (MERT) [Och, 2003], the NIST MT06 as the development test set and the MT08 as the final test set. The numbers of documents/sentences in the NIST MT05, MT06 and MT08 are 100/1082, 79/1664 and 109/1357 respectively. They contain 17, 27.49 and 15.53 lexical chains per document respectively.

We used the case-insensitive BLEU-4 (Papineni

[2] The training data includes LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2004T08 (Hong Kong Hansards/Laws/News).

[3] Available at: http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

ϵ	**MT06**
0.05	30.53
0.1	**31.64**
0.2	31.45
0.3	30.73
0.4	31.01

Table 2: BLEU scores of the probability cohesion model $M_p(T_{D_t}, \{_\epsilon LC_t^k\}_{k=1}^N)$ with different values for the threshold ϵ.

System	**MT06**	**MT08**	**Avg**
Baseline	30.43	23.32	26.88
LexChainCount(top 1)	30.46	23.52	26.99
LexChainCount	30.79	23.34	27.07
LexChainProb	**31.64**	**24.54**	**28.09**

Table 3: Effects of the lexical chain based count and probability cohesion models. LexChainCount: the count model defined in Eq. (3). LexChainProb: the probability model defined in Eq. (5).

et al., 2002) as our evaluation metric. As MERT is normally instable, we ran the tuning process three times for all our experiments and presented the average BLEU scores on the three MERT runs as suggested by Clark et al (2011).

5.2 Setting the Threshold ϵ

As the two lexical chain based cohesion models are built on the super target lexical chains that are associated with a parameter ϵ, we need to tune the threshold parameter ϵ on the development test set NIST MT06. We conducted a group of experiments using the probability cohesion model defined in Eq. (5) to find the best threshold. Experiment results are shown in Table 2.

If we set the threshold too small (e.g., 0.05), the super target lexical chains may contain too many noisy words that are not the translations of source lexical chain words, which may jeopardise the quality of the super target lexical chains. The cohesion model built on these noisy super target lexical chains may select incorrect words rather than the proper lexical chain words. On the other hand, if we set the threshold too large (e.g., 0.3 or 0.4), we may take the risk of not selecting the appropriate chain word translations into the super target lexical chains. It seems that the best threshold is 0.1 as we obtained the highest BLEU score 31.64 on the NIST MT06 with this threshold. Therefore we set the threshold ϵ to 0.1 in all experiments thereafter.

5.3 Effect of the Count and Probability Cohesion Model

After we found the best threshold, we carried out experiments to test the effect of the two lexical chain based cohesion models: the count and probability cohesion model. We compared them against the baseline system that does not integrate any lexical chain information. We also compared the count cohesion model (LexChainCount(top1)) built on the target lexical chains where each target chain word is the best translation of its corresponding source lexical chain word according to Eq. (1). Experiment results are shown in Table 3.

From Table 3, we can observe that

- Our lexical chain based cohesion models are able to substantially improve the translation quality in terms of BLEU score. We achieve an average improvement of up to 1.21 BLEU points over the baseline on the two test sets MT06 and MT08.
- The count cohesion model built on the super target lexical chains is better than that based on the target lexical chains only with top one translations (27.07 vs. 26.99). This shows the advantage of the super target lexical chains $\{_\epsilon LC_t^k\}_{k=1}^N$ over the standard target lexical chians $\{LC_t^k\}_{k=1}^N$.
- Finally, the probability cohesion model is much better than the count cohesion model (28.09 vs. 27.07). This suggests that we should take into account chain word translation probabilities when we reward hypotheses where target lexical chain words occur.

5.4 Lexical Chains vs. Lexical Cohesion Devices

As we have mentioned in Section 2, lexical cohesion devices can be also used to build lexical cohesion models to capture lexical cohesion relations in a text. We therefore want to compare our lexical chain based cohesion models with the lexical cohesion device based cohesion models.

System	MT06	MT08	Avg
Baseline	30.43	23.32	26.88
LexDeviceTrigger	31.35	24.11	27.73
LexChainProb	31.64	24.54	28.09

Table 4: The lexical chain based probability cohesion model (LexChainProb) vs. the lexical cohesion device based trigger model (LexDeviceTrigger).

We re-implemented the mutual information trigger model that is the best lexical cohesion model based on lexical cohesion devices among the three models proposed by Xiong et al. (2013). The mutual information trigger model measures the association strength of two lexical cohesion items x and y in a lexical cohesion relation xRy. In the model, it is required that x occurs in a sentence preceding the sentence where y occurs and that the two items have a lexical cohesion relation such as word repetition, synonym. The model treats x as the trigger and y as the triggered item. The mutual information between the trigger x and the triggered item y estimates how possible y will occur given x is mentioned in a text.

The comparison results are reported in Table 4. Our lexical chain based probability cohesion model outperforms the lexical cohesion device based trigger model by 0.36 BLEU points. The reason for this superiority of our cohesion model over the trigger model may be that the former model captures lexical cohesion relations among sequences of words through lexical chains while the latter model captures lexical cohesion relations only between two related words.

6 Conclusions

We have presented two lexical chain based cohesion models that incorporate the lexical cohesion structure of a text into document-level machine translation. We project the lexical chains of a source document to the corresponding target document by translating each word in each source lexical chain into their counterparts via MaxEnt classifiers. The projected target lexical chains provide a representation of the lexical cohesion structure of the target document that is to be generated. We build two cohesion models based on the projected target lexical chains: a count model that rewards a hypothesis according to the time of occurrence of target lexical chain words in the hypothesis and a probability model that further takes translation probabilities into account when rewarding hypotheses. These two cohesion models are used to constrain word selection for document translation so that the generated document is consistent with the projected lexical cohesion structure.

We have integrated the two proposed cohesion models into a hierarchical phrase-based SMT system. Experiment results on large-scale data validate that

- The lexical chain based cohesion models are able to substantially improve translation quality in terms of BLEU.
- The probability cohesion model is better than the count cohesion model.
- The lexical chain based probability cohesion model is better than the previous mutual information trigger model that adopts lexical cohesion devices to capture lexical cohesion relations between two related words.

As we mentioned in Section 2, cohesion is closely connected to coherence. It provides a surface indicator for coherence identification (Barzilay and Elhadad, 1997). In the future, we would like to use lexical chains to identify coherence and incorporate both cohesion and coherence into document-level machine translation.

References

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Beata Beigman Klebanov and Michael Flor. 2013. Associative texture is lost in translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 27–32, Sofia, Bulgaria, August. Association for Computational Linguistics.

Marine Carpuat and Dekai Wu. 2007a. How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. In *Proceedings of the 11th Conference on Theoretical and*

Methodological Issues in Machine Translation, pages 43–52.

Marine Carpuat and Dekai Wu. 2007b. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic, June. Association for Computational Linguistics.

Marine Carpuat. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 19–27, Boulder, Colorado, June. Association for Computational Linguistics.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40, Prague, Czech Republic, June. Association for Computational Linguistics.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: a chinese language technology platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 13–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June.

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, February.

Michel Galley and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th international joint conference on Artificial intelligence*, IJCAI'03, pages 1486–1488, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919, Edinburgh, Scotland, UK., July.

Liane Guillou. 2013. Analysing lexical consistency in translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 10–18, Sofia, Bulgaria, August. Association for Computational Linguistics.

M.A.K Halliday and Ruqayia Hasan. 1976. *Cohesion in English*. London: Longman.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea, July.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168, Jeju Island, Korea, July. Association for Computational Linguistics.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.*, 17(1):21–48, March.

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*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

M.A. Stairmand. 1996. *A computational analysis of lexical cohesion with applications in information retrieval*. UMIST.

Andreas Stolcke. 2002. Srilm–an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, USA, September.

Jörg Tiedemann. 2010. Context adaptation in statistical machine translation using models with exponentially decaying cache. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15, Uppsala, Sweden, July.

Ferhan Ture, Douglas W. Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426, Montréal, Canada, June.

Muriel Vasconcellos. 1989. Cohesion and coherence in the presentation of machine translation products. In James E.Alatis, editor, *Geogetown University Round Table on Languages and Linguistics 1989*, pages 89–105, Washington, D.C. Georgetown University Press.

Billy T. M. Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the*

2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1060–1068, Jeju Island, Korea, July.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level consistency verification in machine translation. In *Proceedings of the 2011 MT summit XIII*, pages 131–138, Xiamen, China, September.

Deyi Xiong and Min Zhang. 2013. A topic-based coherence model for statistical machine translation. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13)*, Bellevue, Washington, USA, July.

Deyi Xiong, Guosheng Ben, Min Zhang, Yajuan Lv, and Qun Liu. 2013. Modeling lexical cohesion for document-level machine translation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI-13)*, Beijing, China, August.