Translation Acquisition Using Synonym Sets

Daniel Andrade **Masaaki Tsuchida** **Takashi Onishi** **Kai Ishikawa**
Knowledge Discovery Research Laboratories, NEC Corporation, Nara, Japan
{s-andrade@cj, m-tsuchida@cq,
t-onishi@bq, k-ishikawa@dq}.jp.nec.com

Abstract

We propose a new method for translation acquisition which uses a set of synonyms to acquire translations from comparable corpora. The motivation is that, given a certain query term, it is often possible for a user to specify one or more synonyms. Using the resulting set of query terms has the advantage that we can overcome the problem that a single query term's context vector does not always reliably represent a terms meaning due to the context vector's sparsity. Our proposed method uses a weighted average of the synonyms' context vectors, that is derived by inferring the mean vector of the von Mises-Fisher distribution. We evaluate our method, using the synsets from the cross-lingually aligned Japanese and English WordNet. The experiments show that our proposed method significantly improves translation accuracy when compared to a previous method for smoothing context vectors.

1 Introduction

Automatic translation acquisition is an important task for various applications. For example, finding term translations can be used to automatically update existing bilingual dictionaries, which are an indispensable resource for tasks such as cross-lingual information retrieval and text mining.

Various previous research like (Rapp, 1999; Fung, 1998) has shown that it is possible to acquire word translations from comparable corpora.

We suggest here an extension of this approach which uses several query terms instead of a single query term. A user who searches a translation for a query term that is not listed in an existing bilingual dictionary, might first try to find a synonym of that term. For example, the user might look up a synonym in a thesaurus[1] or might use methods for automatic synonym acquisition like described in (Grefenstette, 1994). If the synonym is listed in the bilingual dictionary, we can consider the synonym's translations as the translations of the query term. Otherwise, if the synonym is not listed in the dictionary either, we use the synonym together with the original query term to find a translation.

We claim that using a set of synonymous query terms to find a translation is better than using a single query term. The reason is that a single query term's context vector is, in general, unreliable due to sparsity. For example, a low frequent query term tends to have many zero entries in its context vector. To mitigate this problem it has been proposed to smooth a query's context vector by its nearest neighbors (Pekar et al., 2006). However, nearest neighbors, which context vectors are close the query's context vector, can have different meanings and therefore might introduce noise.

The contributions of this paper are two-fold. First, we confirm experimentally that smoothing a query's context vector with its synonyms leads in deed to higher translation accuracy, compared to smoothing with nearest neighbors. Second, we propose a simple method to combine a set of context vectors that performs in this setting better than a method previously proposed by (Pekar et al., 2006).

Our approach to combine a set of context vec-

[1] Monolingual thesauri are, arguably, easier to construct than bilingual dictionaries.

Proceedings of NAACL-HLT 2013, pages 655–660,
Atlanta, Georgia, 9–14 June 2013.

tors is derived by learning the mean vector of a von Mises-Fisher distribution. The combined context vector is a weighted-average of the original context-vectors, where the weights are determined by the word occurrence frequencies.

In the following section we briefly show the relation to other previous work. In Section 3, we explain our method in detail, followed by an empirical evaluation in Section 4. We summarize our results in Section 6.

2 Related Work

There are several previous works on extracting translations from comparable corpora ranging from (Rapp, 1999; Fung, 1998), and more recently (Haghighi et al., 2008; Laroche and Langlais, 2010), among others. Essentially, all these methods calculate the similarity of a query term's context vector with each translation candidate's context vector. The context vectors are extracted from the comparable corpora, and mapped to a common vector space with the help of an existing bilingual dictionary.

The work in (Déjean et al., 2002) uses cross-lingually aligned classes in a multilingual thesaurus to improve the translation accuracy. Their method uses the probability that the query term and a translation candidate are assigned to the same class. In contrast, our method does not need cross-lingually aligned classes.

Ismail and Manandhar (2010) proposes a method that tries to improve a query's context vector by using in-domain terms. In-domain terms are the terms that are highly associated to the query, as well as highly associated to one of the query's highly associated terms. Their method makes it necessary that the query term has enough highly associated context terms.[2] However, a low-frequent query term might not have enough highly associated terms.

In general if a query term has a low-frequency in the corpus, then its context vector is sparse. In that case, the chance of finding a correct translation is reduced (Pekar et al., 2006). Therefore, Pekar et al. (2006) suggest to use distance-based averaging to smooth the context vector of a low-frequent query term. Their smoothing strategy is dependent on the occurrence frequency of a query term and its close neighbors. Let us denote $\boldsymbol{q}$ the context vector of the query word, and K be the set of its close neighbors. The smoothed context vector $\boldsymbol{q}'$ is then derived by using:

$$\boldsymbol{q}' = \gamma \cdot \boldsymbol{q} + (1-\gamma) \cdot \sum_{\boldsymbol{x} \in K} w_x \cdot \boldsymbol{x}\,, \qquad (1)$$

where w_x is the weight of neighbor x, and all weights sum to one. The context vectors $\boldsymbol{q}$ and $\boldsymbol{x}$ are interpreted as probability vectors and therefore L1-normalized. The weight w_x is a function of the distance between neighbor x and query q. The parameter γ determines the degree of smoothing, and is a function of the frequency of the query term and its neighbors:

$$\gamma = \frac{\log f(q)}{\log \max_{x \in K \cup \{q\}} f(x)} \qquad (2)$$

where $f(x)$ is the frequency of term x. Their method forms the baseline for our proposed method.

3 Proposed Method

Our goal is to combine the context vectors to one context vector which is less sparse and more reliable than the original context vector of query word q. We assume that for each occurrence of a word, its corresponding context vector was generated by a probabilistic model. Furthermore, we assume that synonyms are generated by the same probability distribution. Finally we use the mean vector of that distribution to represent the combined context vector. By using the assumption that *each* occurrence of a word corresponds to one sample of the probability distribution, our model places more weight on synonyms that are highly-frequent than synonyms that occur infrequently. This is motivated by the assumption that context vectors of synonyms that occur with high frequency in the corpus, are more reliable than the ones of low-frequency synonyms.

When comparing context vectors, work like Laroche and Langlais (2010) observed that often the cosine similarity performs superior to other distance-measures, like, for example, the euclidean distance. This suggests that context vectors tend to lie in the spherical vector space,

[2] In their experiments, they require that a query word has at least 100 associated terms.

and therefore the von Mises-Fisher distribution is a natural choice for our probabilistic model. The von Mises-Fisher distribution was also successfully used in the work of (Basu et al., 2004) to cluster text data.

The von Mises-Fisher distribution with location parameter $\boldsymbol{\mu}$, and concentration parameter κ is defined as:

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \kappa) = c(\kappa) \cdot e^{\kappa \cdot \boldsymbol{x} \cdot \boldsymbol{\mu}^T},$$

where $c(\kappa)$ is a normalization constant, and $||\boldsymbol{x}|| = ||\boldsymbol{\mu}|| = 1$, and $\kappa \geq 0$. $||$ denotes here the L2-norm. The cosine-similarity measures the angle between two vectors, and the von Mises distribution defines a probability distribution over the possible angles. The parameter $\boldsymbol{\mu}$ of the von Mises distribution is estimated as follows (Jammalamadaka and Sengupta, 2001): Given the words $x_1, ..., x_n$, we denote the corresponding context vectors as $\mathbf{x}_1, ..., \mathbf{x}_n$, and assume that each context vector is L2-normalized. Then, the mean vector $\boldsymbol{\mu}$ is calculated as:

$$\boldsymbol{\mu} = \frac{1}{Z} \sum_{i=1}^{n} \frac{\mathbf{x}_i}{n}$$

where Z ensures that the resulting context vector is L2-normalized, i.e. Z is $|| \sum_{i=1}^{n} \frac{\mathbf{x}_i}{n} ||$. For our purpose, κ is irrelevant and is assumed to be any fixed positive constant.

Since we assume that each occurrence of a word x in the corpus corresponds to one observation of the corresponding word's context vector $\mathbf{x}$, we get the following formula:

$$\boldsymbol{\mu} = \frac{1}{Z'} \cdot \sum_{i=1}^{n} \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)} \cdot \mathbf{x}_i$$

where Z' is now $|| \sum_{i=1}^{n} \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)} \cdot \mathbf{x}_i ||$. We then use the vector $\boldsymbol{\mu}$ as the combined vector of the words' context vectors $\mathbf{x}_i$.

Our proposed procedure to combine the context vector of query word q and its synonyms can be summarized as follows:

1. Denote the context vectors of q and its synonyms as $\mathbf{x}_1, ..., \mathbf{x}_n$, and L2-normalize each context vector.
2. Calculate the weighted average of the vectors $\mathbf{x}_1, ..., \mathbf{x}_n$, whereas the weights correspond to the frequencies of each word x_i.
3. L2-normalize the weighted average.

4 Experiments

As source and target language corpora we use a corpus extracted from a collection of complaints concerning automobiles compiled by the Japanese Ministry of Land, Infrastructure, Transport and Tourism (MLIT)[3] and the USA National Highway Traffic Safety Administration (NHTSA)[4], respectively. The Japanese corpus contains 24090 sentences that were POS tagged using MeCab (Kudo et al., 2004). The English corpus contains 47613 sentences, that were POS tagged using Stepp Tagger (Tsuruoka et al., 2005), and use the Lemmatizer (Okazaki et al., 2008) to extract and stem content words (nouns, verbs, adjectives, adverbs).

For creating the context vectors, we calculate the association between two content words occurring in the same sentence, using the log-odds-ratio (Evert, 2004). It was shown in (Laroche and Langlais, 2010) that the log-odds-ratio in combination with the cosine-similarity performs superior to several other methods like PMI[5] and LLR[6]. For comparing two context vectors we use the cosine similarity.

To transform the Japanese and English context vectors into the same vector space, we use a bilingual dictionary with around 1.6 million entries.[7] To express all context vectors in the same vector space, we map the context vectors in English to context vectors in Japanese.[8] First, for all the words which are listed in the bilingual dictionary we calculate word translation probabilities. These translation probabilities are calculated using the EM-algorithm described in (Koehn and Knight, 2000). We then create a translation matrix T which contains in each

[3] http://www.mlit.go.jp/jidosha/carinf/rcl/defects.html

[4] http://www-odi.nhtsa.dot.gov/downloads/index.cfm

[5] point-wise mutual information

[6] log-likelihood ratio

[7] The bilingual dictionary was developed in the course of our Japanese language processing efforts described in (Sato et al., 2003).

[8] Alternatively, we could, for example, use canonical correlation analysis to match the vectors to a common latent vector space, like described in (Haghighi et al., 2008).

column the translation probabilities for a word in English into any word in Japanese. Each context vector in English is then mapped into Japanese using the linear transformation described by the translation matrix T. For word x with context vector $\mathbf{x}$ in English, let $\mathbf{x}'$ be its context vector after transformation into Japanese, i.e. $\mathbf{x}' = T \cdot \mathbf{x}$.

The gold-standard was created by considering all nouns in the Japanese and English WordNet where synsets are aligned cross-lingually. This way we were able to create a gold-standard with 215 Japanese nouns, and their respective English translations that occur in our comparable corpora.[9] Note that the cross-lingual alignment is needed only for evaluation. For evaluation, we consider only the translations that occur in the corresponding English synset as correct.

Because all methods return a ranked list of translation candidates, the accuracy is measured using the rank of the translation listed in the gold-standard. The inverse rank is the sum of the inverse ranks of each translation in the gold-standard.

In Table 1, the first row shows the results when using no smoothing. Next, we smooth the query's context vector by using Equation (1) and (2). The set of neighbors K is defined as the k-terms in the source language that are closest to the query word, with respect to the cosine similarity (sim). The weight w_x for a neighbor x is set to $w_x = 10^{0.13 \cdot sim(x,q)}$ in accordance to (Pekar et al., 2006). For k we tried values between 1 and 100, and got the best inverse rank when using k=19. The resulting method (Top-k Smoothing) performs consistently better than the method using no smoothing, see Table 1, second row. Next, instead of smoothing the query word with its nearest neighbors, we use as the set K the set of synonyms of the query word (Syn Smoothing). Table 1 shows a clear improvement over the method that uses nearest neighbor-smoothing. This confirms our claim that using synonyms for smoothing can lead to better translation accuracy than using nearest neighbors. In the last row of Table 1, we compare our proposed method to combine context vectors of synonyms (Syn Mises-Combination), with the previous method (Syn Smoothing). A pair-wise comparison of our proposed method with Syn Smoothing shows a statistically significant improvement ($p < 0.01$).[10]

Finally, we also show the result when simply adding each synonym vector to the query's context vector to form a new combined context vector (Syn Sum).[11] Even though, this approach does not use the frequency information of a word, it performs better than Syn Smoothing. We suppose that this is due to the fact that it actually indirectly uses frequency information, since the log-odds-ratio tends to be higher for words which occur with high frequency in the corpus.

Method	Top1	Top5	Top10	MIR
No Smoothing	0.14	0.30	0.36	0.23
Top-k Smoothing	0.16	0.33	0.43	0.26
Syn Smoothing	0.18	0.35	0.46	0.28
Syn Sum	0.23	0.46	0.57	0.35
Syn Mises-Combination	0.31	0.46	0.55	0.40

Table 1: Shows Top-n accuracy and mean inverse rank (MIR) for baseline methods which use no synonyms (No Smoothing, Top-k Smoothing), the proposed method (Syn Mises-Combination) which uses synonyms, and alternative methods that also use synonyms (Syn Smoothing, Syn Sum).

5 Discussion

We first discuss an example where the query terms are クルーズ (cruise) and 巡航 (cruise). Both words can have the same meaning. The resulting translation candidates suggested by the baseline methods and the proposed method is shown in Table 2. Using no smoothing, the baseline method outputs the correct translation for クルーズ (cruise) and 巡航 (cruise) at rank 10 and 15, respectively. When combining both queries to form one context vector our proposed method (Syn Mises-Combination) retrieves the correct translation at rank 2. Note that we considered all nouns that occur three or more times as possible translation candidates. As can be seen in Table 2, this also includes spelling mistakes like "sevice" and "infromation".

[9] The resulting synsets in Japanese and English, contain in average 2.2 and 2.8 words, respectively. The ambiguity of a query term in our gold-standard is low, since, in average, a query term belongs to only 1.2 different synsets.

[10] We use the sign-test (Wilcox, 2009) to test the hypothesis that the proposed method ranks higher than the baseline.

[11] No normalization is performed before adding the context vectors.

Method	Query	Output	Rank
No Smoothing	クルーズ	..., affinity, delco, **cruise**, sevice, sentrum,...	10
No Smoothing	巡航	..., denali, attendant, **cruise**, abs, tactic,...	15
Top-k Smoothing	クルーズ	pillar, multi, **cruise**, star, affinity,...	3
Top-k Smoothing	巡航	..., burnout, dipstick, **cruise**, infromation, speed, ...	8
Syn Smoothing	クルーズ smoothed with 巡航	..., affinity, delco, **cruise**, sevice, sentrum,...	10
Syn Smoothing	巡航 smoothed with クルーズ	..., alldata, mode, **cruise**, expectancy, mph,...	8
Syn Sum	クルーズ, 巡航	assumption, level, **cruise**, reimbursment, infromation,...	3
Syn Mises-Combination	クルーズ, 巡航	pillar, **cruise**, assumption, level, speed,...	2

Table 2: Shows the results for クルーズ and 巡航 which both have the same meaning "cruise". The third column shows part of the ranked translation candidates separated by comma. The last column shows the rank of the correct translation "cruise". Syn Smoothing uses Equation (1) with $\mathbf{q}$ corresponding to the context vector of the query word, and K contains only the context vector of the term that is used for smoothing.

Finally, we note that some terms in our test set are ambiguous, and the ambiguity is not resolved by using the synonyms of only one synset. For example, the term 操舵 (steering, guidance) belongs to the synset "steering, guidance" which includes the terms 舵取り (steering, guidance) and ガイド (guidance), 案内 (guidance). Despite this conflation of senses in one synset, our proposed method can improve the finding of (one) correct translation. The baseline system using only 操舵 (steering, guidance) outputs the correct translation "steering" at rank 4, whereas our method using all four terms outputs it at rank 2.

6 Conclusions

We proposed a new method for translation acquisition which uses a set of synonyms to acquire translations. Our approach combines the query term's context vector with all the context vectors of its synonyms. In order to combine the vectors we use a weighted average of each context vector, where the weights are determined by a term's occurrence frequency.

Our experiments, using the Japanese and English WordNet (Bond et al., 2009; Fellbaum, 1998), show that our proposed method can increase the translation accuracy, when compared to using only a single query term, or smoothing with nearest neighbours. Our results suggest that instead of directly searching for a translation, it is worth first looking for synonyms, for example by considering spelling variations or monolingual resources.

References

S. Basu, M. Bilenko, and R.J. Mooney. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68.

F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki. 2009. Enhancing the japanese wordnet. In *Proceedings of the 7th Workshop on Asian Language Resources*, pages 1–8. Association for Computational Linguistics.

H. Déjean, É. Gaussier, and F. Sadat. 2002. An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *Proceedings of the International Conference on Computational Linguistics*, pages 1–7. International Committee on Computational Linguistics.

S. Evert. 2004. The statistics of word cooccurrences: word pairs and collocations. *Doctoral dissertation, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart*.

C. Fellbaum. 1998. Wordnet: an electronic lexical database. *Cambrige, MIT Press, Language, Speech, and Communication*.

P. Fung. 1998. A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. *Lecture Notes in Computer Science*, 1529:1–17.

G. Grefenstette. 1994. *Explorations in automatic thesaurus discovery*. Springer.

A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 771–779. Association for Computational Linguistics.

A. Ismail and S. Manandhar. 2010. Bilingual lexicon extraction from comparable corpora using in-domain terms. In *Proceedings of the International Conference on Computational Linguistics*, pages 481 – 489.

S.R. Jammalamadaka and A. Sengupta. 2001. *Topics in circular statistics*, volume 5. World Scientific Pub Co Inc.

P. Koehn and K. Knight. 2000. Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, pages 711–715. Association for the Advancement of Artificial Intelligence.

T. Kudo, K. Yamamoto, and Y. Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 230–237. Association for Computational Linguistics.

A. Laroche and P. Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *Proceedings of the International Conference on Computational Linguistics*, pages 617 – 625.

N. Okazaki, Y. Tsuruoka, S. Ananiadou, and J. Tsujii. 2008. A discriminative candidate generator for string transformations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 447–456. Association for Computational Linguistics.

V. Pekar, R. Mitkov, D. Blagoev, and A. Mulloni. 2006. Finding translations for low-frequency words in comparable corpora. *Machine Translation*, 20(4):247–266.

R. Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 519–526. Association for Computational Linguistics.

K. Sato, T. Ikeda, T. Nakata, and S. Osada. 2003. Introduction of a Japanese language processing middleware used for CRM. In *Annual Meeting of the Japanese Association for Natural Language Processing (in Japanese)*, pages 109–112.

Y. Tsuruoka, Y. Tateishi, J. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. *Lecture Notes in Computer Science*, 3746:382–392.

R.R. Wilcox. 2009. *Basic Statistics: Understanding Conventional Methods and Modern Insights*. Oxford University Press.