Learning the Optimal use of Dependency-parsing Information for Finding Translations with Comparable Corpora

Daniel Andrade[†], **Takuya Matsuzaki**[†], **Jun'ichi Tsujii**[‡]
[†]Department of Computer Science, University of Tokyo
{daniel.andrade, matuzaki}@is.s.u-tokyo.ac.jp
[‡]Microsoft Research Asia, Beijing
jtsujii@microsoft.com

Abstract

Using comparable corpora to find new word translations is a promising approach for extending bilingual dictionaries (semi-) automatically. The basic idea is based on the assumption that similar words have similar contexts across languages. The context of a word is often summarized by using the bag-of-words in the sentence, or by using the words which are in a certain dependency position, e.g. the predecessors and successors. These different context positions are then combined into one context vector and compared across languages. However, previous research makes the (implicit) assumption that these different context positions should be weighted as equally important. Furthermore, only the same context positions are compared with each other, for example the successor position in Spanish is compared with the successor position in English. However, this is not necessarily always appropriate for languages like Japanese and English. To overcome these limitations, we suggest to perform a linear transformation of the context vectors, which is defined by a matrix. We define the optimal transformation matrix by using a Bayesian probabilistic model, and show that it is feasible to find an approximate solution using Markov chain Monte Carlo methods. Our experiments demonstrate that our proposed method constantly improves translation accuracy.

1 Introduction

Using comparable corpora to automatically extend bilingual dictionaries is becoming increasingly popular (Laroche and Langlais, 2010; Andrade et al., 2010; Ismail and Manandhar, 2010; Laws et al., 2010; Garera et al., 2009). The general idea is based on the assumption that similar words have similar contexts across languages. The context of a word can be described by the sentence in which it occurs (Laroche and Langlais, 2010) or a surrounding word-window (Rapp, 1999; Haghighi et al., 2008). A few previous studies, like (Garera et al., 2009), suggested to use the predecessor and successors from the dependency-parse tree, instead of a word window. In (Andrade et al., 2011), we showed that including dependency-parse tree context positions together with a sentence bag-of-words context can improve word translation accuracy. However previous works do not make an attempt to find an *optimal* combination of these different context positions.

Our study tries to find an optimal weighting and aggregation of these context positions by learning a linear transformation of the context vectors. The motivation is that different context positions might be of different importance, e.g. the direct predecessors and successors from the dependency tree might be more important than the larger context from the whole sentence. Another motivation is that dependency positions cannot be always compared across different languages, e.g. a word which tends to occur as a modifier in English, can tend to occur in Japanese in a different dependency position.

As a solution, we propose to learn the optimal combination of dependency and bag-of-words sentence information. Our approach uses a linear transformation of the context vectors, before comparing

Proceedings of the 4th Workshop on Building and Using Comparable Corpora, pages 10–18,
49th Annual Meeting of the Association for Computational Linguistics,
Portland, Oregon, 24 June 2011. ©*2011 Association for Computational Linguistics*

them using the cosine similarity. This can be considered as a generalization of the cosine similarity. We define the optimal transformation matrix by the maximum-a-posterior (MAP) solution of a Bayesian probabilistic model. The likelihood function for a translation matrix is defined by considering the expected achieved translation accuracy. As a prior, we use a Dirichlet distribution over the diagonal elements in the matrix and a uniform distribution over its non-diagonal elements. We show that it is feasible to find an approximation of the optimal solution using Markov chain Monte Carlo (MCMC) methods. In our experiments, we compare the proposed method, which uses this approximation, with the baseline method which uses the cosine similarity without any linear transformation. Our experiments show that the translation accuracy is constantly improved by the proposed method.

In the next section, we briefly summarize the most relevant previous work. In Section 3, we then explain the baseline method which is based on previous research. Section 4 explains in detail our proposed method, followed by Section 5 which provides an empirical comparison to the baseline, and analysis. We summarize our findings in Section 6.

2 Previous Work

Using comparable corpora to find new translations was pioneered in (Rapp, 1999; Fung, 1998). The basic idea for finding a translation for a word q (query), is to measure the context of q and then to compare the context with each possible translation candidate, using an existing dictionary. We will call words for which we have a translation in the given dictionary, *pivot* words. First, using the source corpus, they calculate the degree of association of a query word q with all pivot words. The degree of association is a measure which is based on the co-occurrence frequency of q and the pivot word in a certain context position. A context (position) can be a word-window (Rapp, 1999), sentence (Utsuro et al., 2003), or a certain position in the dependency-parse tree (Garera et al., 2009; Andrade et al., 2011). In this way, they get a context vector for q, which contains the degree of association to the pivot words in different context positions. Using the target corpus, they then calculate a context vector for each possible translation candidate x, in the same way. Finally, they compare the context vector of q with the context vector of each candidate x, and retrieve a ranked list of possible translation candidates. In the next section, we explain the baseline which is based on that previous research.

The general idea of *learning* an appropriate method to compare high-dimensional vectors is not new. Related research is often called "metric-learning", see for example (Xing et al., 2003; Basu et al., 2004). However, for our objective function it is difficult to find an analytic solution. To our knowledge, the idea of parameterizing the transformation matrix, in the way we suggest in Section 4, and to learn an approximate solution with a fast sampling strategy is new.

3 Baseline

Our baseline measures the degree of association between the query word q and each pivot word with respect to several context positions. As a context position we consider the predecessors, successors, siblings with respect to the dependency parse tree, and the whole sentence (bag-of-words). The dependency information which is used is also illustrated in Figure 1. As a measure of the degree of association we use the Log-odds-ratio as proposed in (Laroche and Langlais, 2010).

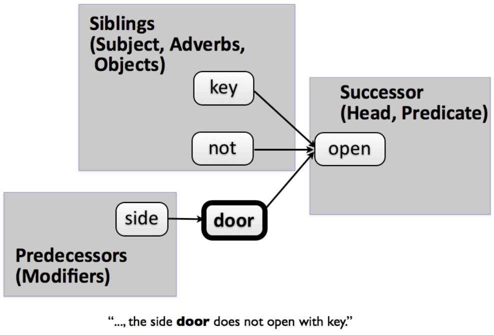

Figure 1: Example of the dependency information used by our approach. Here, from the perspective of "door".

Next, we define the context vector which contains the degree of association between the query and each pivot in several context positions. First, for each context position i we define a vector $\mathbf{q_i}$ which contains the degree of association with each pivot word in the context position i. If we number the pivot words from 1 to n, then this vector can be written as $\mathbf{q_i} = (q_i^1, \ldots, q_i^n)$. Note that in our case i ranges from 1 to 4, representing the context positions predecessors (1), successors (2), siblings (3), and the sentence bag-of-words (4). Finally, the complete context vector for the query q is a long vector $\mathbf{q}$ which appends each $\mathbf{q_i}$, i.e.: $\mathbf{q} = (\mathbf{q_1}, \ldots, \mathbf{q_4})$. Next, in the same way as before, we create a context vector $\mathbf{x}$ for each translation candidate x in the target language. For simplicity, we assume that each pivot word in the source language has only one corresponding translation in the target language. As a consequence, the dimensions of $\mathbf{q}$ and $\mathbf{x}$ are the same. Finally we can score each translation candidate by using the cosine similarity between $\mathbf{q}$ and $\mathbf{x}$.

We claim that all of the context positions (1 to 4) can contain information which is helpful to identify translation candidates. However, we do not know about their relative importance, neither do we know whether these dependency positions can be compared across language pairs as different as Japanese and English. The cosine similarity simply weights all dependency position equally important and ignores problems which might occur when comparing dependency positions across languages.

4 Proposed Method

Our proposed method tries to overcome the shortcomings of the cosine-similarity by using the following generalization:

$$sim(\mathbf{q}, \mathbf{x}) = \frac{\mathbf{q}\mathbf{A}\mathbf{x}^T}{\sqrt{\mathbf{q}\mathbf{A}\mathbf{q}^T}\sqrt{\mathbf{x}\mathbf{A}\mathbf{x}^T}}\,, \tag{1}$$

where A is a positive-definite matrix in $\mathbb{R}^{dn \times dn}$, and ^{T} is the transpose of a vector. This can also be considered as linear transformation of the vectors using $\sqrt{\mathbf{A}}$ before using the normal cosine similarity, see also (Basu et al., 2004).[1]

The challenge is to find an appropriate matrix A which is expected to take the correlations between the different dimensions into account, and which optimally weights the different dimensions. Note that, if we set A to the identity matrix, we recover the normal cosine similarity, which is our baseline.

Clearly, finding an optimal matrix in $\mathbb{R}^{dn \times dn}$ is infeasible due to the high dimensionality. We will therefore restrict the structure of A.

Let $\mathbf{I}$ be the identity matrix in $\mathbb{R}^{n \times n}$, then we define the matrix A, as follows:

$$\mathbf{A} = \begin{pmatrix} d_1\mathbf{I} & z_{1,2}\mathbf{I} & z_{1,3}\mathbf{I} & z_{1,4}\mathbf{I} \\ z_{1,2}\mathbf{I} & d_2\mathbf{I} & z_{2,3}\mathbf{I} & z_{2,4}\mathbf{I} \\ z_{1,3}\mathbf{I} & z_{2,3}\mathbf{I} & d_3\mathbf{I} & z_{3,4}\mathbf{I} \\ z_{1,4}\mathbf{I} & z_{2,4}\mathbf{I} & z_{3,4}\mathbf{I} & d_4\mathbf{I} \end{pmatrix}$$

It is clear from this definition that $d_1, \ldots, d_4$ weights the context positions 1 to 4. Furthermore, $z_{i,j}$ can be interpreted as a the confusion coefficient between context position i and j. For example, a high value for $z_{2,3}$ means that a pivot word which occurs in the sibling position in Japanese (source language), might not necessarily occur in the sibling position in English (target language), but instead in the successor position. However, in order to reduce the dimensionality of the parameter space further, we assume that each such $z_{i,j}$ has the same value z. Therefore, matrix A becomes

$$\mathbf{A} = \begin{pmatrix} d_1\mathbf{I} & z\mathbf{I} & z\mathbf{I} & z\mathbf{I} \\ z\mathbf{I} & d_2\mathbf{I} & z\mathbf{I} & z\mathbf{I} \\ z\mathbf{I} & z\mathbf{I} & d_3\mathbf{I} & z\mathbf{I} \\ z\mathbf{I} & z\mathbf{I} & z\mathbf{I} & d_4\mathbf{I} \end{pmatrix}.$$

In the next subsection we will explain how we define an optimal solution for A.

4.1 Optimal solution for A

We use a Bayesian probabilistic model in order to define the optimal solution for A. Formally we try to find the maximum-a-posterior (MAP) solution of A, i.e.:

$$\arg\max_A p(A|data, \alpha). \tag{2}$$

The posterior probability is defined by

$$p(A|data, \alpha) \propto f_{auc}(data|A) \cdot p(A|\alpha)\,. \tag{3}$$

$f_{auc}(data|A)$ is the (unnormalized) likelihood function. $p(A|\alpha)$ is the prior that captures our prior beliefs about A, and which is parameterized by a hyperparameter α.

[1] Therefore, exactly speaking A is not the transformation matrix, however it defines uniquely the transformation matrix $\sqrt{\mathbf{A}}$.

4.1.1 The likelihood function $f_{auc}(data|A)$

As a likelihood function we use a modification of the area under the curve (AUC) of the accuracy-vs-rank graph. The accuracy-vs-rank graph shows the translation accuracy at different ranks. $data$ refers to the part of the gold-standard which is used for training. Our complete gold-standard contains 443 domain-specific Japanese nouns (query words). Each Japanese noun in the gold standard corresponds to one pair of the form <Japanese noun (query), English translations (answers)>. We denote the accuracy at rank r, by acc_r. The accuracy acc_r is determined by counting how often the correct answer is listed in the top r translation candidates suggested for a query, divided by the number of all queries in $data$. The likelihood function is now defined as follows:

$$f_{auc}(data|A) = \sum_{r=1}^{20} acc_r \cdot (21 - r)\,. \qquad (4)$$

That means $f_{auc}(data|A)$ accumulates the accuracies at the ranks from 1 to 20, where we weight accuracies at top ranks higher.

4.1.2 The prior $p(A|\alpha)$

The prior over the transformation matrix is factorized in the following manner:

$$p(A|\alpha) = p(z|d_1, \ldots, d_4) \cdot p(d_1, \ldots, d_4|\alpha)\,.$$

The prior over the diagonal is defined as a Dirichlet distribution:

$$p(d_1, \ldots, d_4|\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{4} d_i^{\alpha-1}$$

where α is the concentration parameter of the symmetric Dirichlet, and $B(\alpha)$ is the normalization constant. The prior over the non-diagonal value a is defined as:

$$p(z|d_1, \ldots, d_4) = \frac{1}{\lambda} \cdot 1_{[0,\lambda]}(z) \qquad (5)$$

where $\lambda = min\{d_1, \ldots, d_4\}$.

First, note that our prior limits the possible matrices A to matrices which have diagonal entries which are between 0 and 1. This is not a restriction since the ranking of the translation candidates induced by the parameterized cosine similarity will not change if A is multiplied by a constant $c > 0$. To see this, note that

$$sim(\mathbf{q}, \mathbf{x}) = \frac{\mathbf{q}(c \cdot \mathbf{A})\mathbf{x}}{\sqrt{\mathbf{q}(c \cdot \mathbf{A})\mathbf{q}}\sqrt{\mathbf{x}(c \cdot \mathbf{A})\mathbf{x}}} = \frac{\mathbf{qAx}}{\sqrt{\mathbf{qAq}}\sqrt{\mathbf{xAx}}}\,.$$

Second, note that our prior limits A further, by requiring, in Equation (5), that every non-diagonal element is smaller or equal than any diagonal element. That requirement is sensible since we do not expect that a optimal similarity measure between English and Japanese will prefer context which is similar in *different* dependency positions, over context which is similar in the *same* context positions. To see this, imagine the extreme case where for example d_1 is 0, and instead z_{12} is 1. In that case the similarity measure would ignore any similarity in the predecessor position, but would instead compare the predecessors in Japanese with the successors in English.

Finally, note that our prior puts probability mass over a subset of the *positive-definite* matrices in $\mathbb{R}^{4\times4}$, and puts no probability mass on matrices which are not positive-definite. As a consequence, the similarity measure in Equation (1) is ensured to be well-defined.

4.2 Training

In the following we explain how we use the training data in order to find a good solution for the matrix A.

4.2.1 Setting hyperparameter α

Recall, that α weights our prior belief about how strong we think that the different context positions should be weighted equally. From a practical point-of-view, we do not know how strong we should weight that prior belief. We therefore use empirical Bayes to estimate α, that is we use part of the training data to set α. First, using half of the training set, we find the A which maximizes $p(A|data, \alpha)$ for several α. Then, the remaining half of the training set is used to evaluate $f_{auc}(data|A)$ to find the best α. Note that the prior $p(A|\alpha)$ can also be considered as a regularization to prevent overfitting. In the next sub-section we will explain how to find an approximation of A which maximizes $p(A|data, \alpha)$.

4.2.2 Finding a MAP solution for A

Recall that matrix A is defined by using only five parameters. Since the problem is low-dimensional, we can therefore expect to find a reasonable solution using sampling methods. For finding an approximation of the maximum-a-posteriori (MAP) solution of $p(A|data, \alpha)$, we use the following Markov chain Monte Carlo procedure:

1. Initialize $d_1, \ldots, d_4$ and z.

2. Leave z constant, and run Simulated-Annealing to find the $d_1, \ldots, d_4$ which maximize $p(A|data, \alpha)$.

3. Given $d_1, \ldots, d_4$, sample from the uniform distribution $[1, \min(d_1, \ldots d_4)]$ in order to find the z which maximizes $p(A|data, \alpha)$.

The steps 2. and 3. are repeated till the convergence of the parameters.

Concerning step 2., we use Simulated-Annealing for finding a (local) maximum of $p(d_1, \ldots, d_4|data, \alpha)$ with the following settings: As a jumping distribution we use a Dirichlet distribution which we update every 1000 iterations. The cooling rate is set to $\frac{1}{iteration}$.

For step 2. and 3. it is of utmost importance to be able to evaluate $p(A|data, \alpha)$ fast. The computationally expensive part of $p(A|data, \alpha)$ is to evaluate $f_{auc}(data|A)$. In order to quickly evaluate $f_{auc}(data|A)$, we need to pre-calculate part of $sim(q, x)$ for all queries q and all translation candidates x. To illustrate the basic idea, consider $sim(q, x)$ without the normalization of $\mathbf{q}$ and $\mathbf{x}$ with respect to A, i.e.:

$$sim(q, x) = \mathbf{q}\mathbf{A}\mathbf{x}^T = (\mathbf{q_1}, \ldots, \mathbf{q_4})\mathbf{A}(\mathbf{x_1}, \ldots, \mathbf{x_4})^T .$$

Let us denote $\mathbf{I}^-_{dn}$ a block matrix in $\mathbb{R}^{dn \times dn}$ which contains in each $n \times n$ block the identity matrix except in its diagonal; the diagonal of $\mathbf{I}^-_{dn}$ contains the $n \times n$ matrix which is zero in all entries. We can now rewrite matrix A as:

$$A = \begin{pmatrix} d_1\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & d_2\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & d_3\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & d_4\mathbf{I} \end{pmatrix} + z \cdot \mathbf{I}^-_{dn} .$$

And finally we can factor out the parameters $(d_1, \ldots d_4)$ and z in the following way:

$$sim(q, x) = (d_1, \ldots, d_4) \cdot \begin{pmatrix} \mathbf{q}_1\mathbf{x}_1^T \\ \vdots \\ \mathbf{q}_4\mathbf{x}_4^T \end{pmatrix} + z \cdot (\mathbf{q}\mathbf{I}^-_{dn}\mathbf{x}^T)$$

By pre-calculating $\begin{pmatrix} \mathbf{q}_1\mathbf{x}_1^T \\ \vdots \\ \mathbf{q}_4\mathbf{x}_4^T \end{pmatrix}$ and $\mathbf{q}\mathbf{I}^-_{dn}\mathbf{x}^T$, we can make the evaluation of each sample, in steps 2. and 3., computationally feasible.

5 Experiments

In the experiments of the present study, we used a collection of complaints concerning automobiles compiled by the Japanese Ministry of Land, Infrastructure, Transport and Tourism (MLIT)[2] and another collection of complaints concerning automobiles compiled by the USA National Highway Traffic Safety Administration (NHTSA)[3]. Both corpora are publicly available. The corpora are non-parallel, but are comparable in terms of content. The part of MLIT and NHTSA which we used for our experiments, contains 24090 and 47613 sentences, respectively. The Japanese MLIT corpus was morphologically analyzed and dependency parsed using Juman and KNP[4]. The English corpus NHTSA was POS-tagged and stemmed with Stepp Tagger (Tsuruoka et al., 2005; Okazaki et al., 2008) and dependency parsed using the MST parser (McDonald et al., 2005). Using the Japanese-English dictionary JMDic[5], we found 1796 content words in Japanese which have a translation which is in the English corpus. These content words and their translations correspond to our pivot words in Japanese and English, respectively.[6]

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

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

[4] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/juman.html and http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/knp.html

[5] http://www.csse.monash.edu.au/ jwb/edict_doc.html

[6] Recall that we assume a one-to-one correspondence between a pivot in Japanese and English. If a Japanese pivot word as more than one English translation, we select the translation for which the relative frequency in the target corpus is closest to the pivot in the source corpus.

5.1 Evaluation

For the evaluation we extract a gold-standard which contains Japanese and English noun pairs that actually occur in both corpora.[7] The gold-standard is created with the help of the JMDic dictionary, whereas we correct apparently inappropriate translations, and remove general nouns such as 可能性 (possibility) and ambiguous words such as 米 (rice, America). In this way, we obtain a final list of 443 domain-specific Japanese nouns.

Each Japanese noun in the gold-standard corresponds to one pair of the form <Japanese noun (query), English translations (answers)>. We divide the gold-standard into two halves. The first half is used for for learning the matrix A, the second part is used for the evaluation. In general, we expect that the optimal transformation matrix A depends mainly on the languages (Japanese and English) and on the corpora (MLIT and NHTSA). However, in practice, the optimal matrix can also vary depending on the part of the gold-standard which is used for training. These random variations are especially large, if the part of the gold-standard which is used for training or testing is small.

In order to take these random effects into account, we perform repeated subsampling of the gold-standard. In detail, we randomly split the gold-standard into equally-sized training and test set. This is repeated five times, leading to five training and five test sets. The performance on each test set is shown in Table 1. OPTIMIZED-ALL marks the result of our proposed method, where matrix A is optimized using the training set. The optimization of the diagonal elements $d_1, \ldots, d_4$, and the non-diagonal value z is as described in Section 4.2. Finally, the baseline method, as described in 3, corresponds to OPTIMIZED-ALL where $d_1, \ldots, d_4$ are set to 1, and z is set to 0. This baseline is denoted as NORMAL. We can see that the overall translation accuracy varies across the test sets. However, we see that in all test sets our proposed method OPTIMIZED-ALL performs better than the baseline NORMAL.

5.2 Analysis

In the previous section, we showed that the cosine-similarity is sub-optimal for comparing context vectors which contain information from different context positions. We showed that it is possible to find an approximation of a matrix A which optimally weights, and combines the different context positions. Recall, that the matrix A is described by the parameters $d_1 \ldots d_4$ and z, which can interpreted as context position weights and a confusion coefficient, respectively. Therefore, by looking at these parameters which we learned using each training set, we can get some interesting insights. Table 2 shows theses parameters learned for each training set.

We can see that the parameters, across the training sets, are not as stable as we wish. For example the weight for the predecessor position ranges from 0.27 to 0.44. As a consequence, the average values, shown in the last row of Table 2, have to be interpreted with care. We expect that the variance is due to the limited size of the training set, 220 <query, answers> pairs.

Nevertheless, we can draw some conclusions with confidence. For example, we see that the predecessor and successor positions are the most important contexts, since the weights for both are always higher than for the other context positions. Furthermore, we clearly see that the sibling and sentence (bag-of-words) contexts, although not as highly weighted as the former two, can be considered to be relevant, since each has a weight of around 0.20. Finally, we see that z, the confusion coefficient, is around 0.03, which is small.[8] Therefore, we verify z's usefulness with another experiment. We additionally define the method OPTIMIZED-DIAG which uses the same matrix as OPTIMIZED-ALL except that the confusion coefficient z is set to zero. In Table 1, we can see that the accuracy of OPTIMIZED-DIAG is constantly lower than OPTIMIZED-ALL.

Furthermore, we are interested in the role of the whole sentence (bag-of-words) information which is in the context vector (in position d_4 of the block vector). Therefore, we excluded the sentence informa-

[7] Note that if the current query (Japanese noun) is a pivot word, then the word is not considered as a pivot word.

[8] In other words, z is around 17% of its maximal possible value. The maximal possible value is around 0.18, since, recall that z is, by definition, smaller or equal to $\min\{d_1 \ldots d_4\}$.

Test Set	Method	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy	Top-15 Accuracy	Top-20 Accuracy
1	OPTIMIZED-ALL	**0.20**	**0.37**	**0.47**	**0.50**	**0.54**
	OPTIMIZED-DIAG	**0.20**	0.34	0.43	0.48	0.51
	NORMAL	0.18	0.32	0.43	0.47	0.50
2	OPTIMIZED-ALL	**0.20**	**0.35**	**0.43**	**0.48**	**0.52**
	OPTIMIZED-DIAG	0.19	0.33	0.42	0.46	**0.52**
	NORMAL	0.18	0.34	0.42	0.47	0.49
3	OPTIMIZED-ALL	**0.17**	**0.31**	**0.37**	**0.44**	**0.48**
	OPTIMIZED-DIAG	**0.17**	0.27	0.36	0.41	0.45
	NORMAL	0.16	0.27	0.36	0.41	0.44
4	OPTIMIZED-ALL	0.14	**0.30**	**0.38**	**0.43**	**0.46**
	OPTIMIZED-DIAG	0.14	0.26	0.34	0.4	0.43
	NORMAL	**0.15**	0.29	0.37	0.41	0.44
5	OPTIMIZED-ALL	0.18	**0.34**	**0.42**	**0.46**	**0.51**
	OPTIMIZED-DIAG	0.17	0.30	0.38	0.43	0.48
	NORMAL	**0.19**	0.31	0.40	0.44	0.48
average	OPTIMIZED-ALL	**0.18**	**0.33**	**0.41**	**0.46**	**0.50**
	OPTIMIZED-DIAG	0.17	0.30	0.39	0.44	0.48
	NORMAL	0.17	0.31	0.40	0.44	0.47

Table 1: Shows the accuracy at different ranks for all test sets, and, in the last column, the average over all test sets. The proposed method OPTIMIZED-ALL is compared to the baseline NORMAL. Furthermore, for analysis, the results when optimizing only the diagonal are marked as OPTIMIZED-DIAG.

Training Set	d_1 predecessor	d_2 successor	d_3 sibling	d_4 sentence	z confusion coefficient
1	0.35	0.26	0.19	0.20	0.03
2	0.27	0.29	0.21	0.23	0.03
3	0.35	0.31	0.16	0.18	0.02
4	0.44	0.24	0.17	0.16	0.04
5	0.39	0.28	0.20	0.13	0.03
average	0.36	0.28	0.19	0.18	0.03

Table 2: Shows the parameters which were learned using each training set. $d_1 \ldots d_4$ are the weights of the context positions, which sum up to 1. z marks the degree to which it is useful to compare context across different positions.

tion from the context vector. The accuracy results, averaged over the same test sets as before, are shown in Table 3. We can see that the accuracies are clearly lower than before (compare to Table 1). This clearly justifies to include additionally sentence information into the context vector. It is also interesting to note that the average z value is now 0.14.[9] This is considerable higher than before, and shows that a bag-of-words model can partly make the use of z redundant. However, note that the sentence bag-of-words model covers a broader context, beyond the direct predecessors, successor and siblings, which explains why a small z value is still relevant in the situation where we include sentence bag-of-words into the context vector.

Finally, to see why it can be helpful to compare *different* dependency positions from the context vectors of Japanese and English, we looked at concrete examples. We found, for example, that the translation accuracy of the query word ディスク (disc) improved when using OPTIMIZED-ALL instead of OPTIMIZED-DIAG. The pivot word 歪み (wrap) tends together with both the Japanese query ディスク (disc), and with the correct translation ”disc” in English. However, that pivot word occurs in Japanese and English in different context positions. In the Japanese corpus 歪み (wrap) tends to occur

[9] That is 48% of its maximal possible value. Since for the dependency positions predecessor, successor and sibling we get the average weights 0.38, 0.33 and 0.29, respectively.

Method	Top-1	Top-5	Top-10	Top-15	Top-20
OPT-DEP	**0.13**	**0.25**	**0.34**	**0.38**	**0.41**
NOR-DEP	0.12	0.23	0.29	0.33	0.38

Table 3: The proposed method, but without the sentence information in the context vector, is denoted OPT-DEP. The baseline method, but without the sentence information in the context vector, is denoted NOR-DEP.

together with the query ディスク (disc) in sentences like for example the following:

> "ブレーキ (break) **ディスク (disc)** に歪み (wrap) が生じた (occured)。"

That Japanese sentence can be literally translated as "A wrap occured in the brake disc.", where "wrap" is the sibling of "disc" in the dependency tree. However, in English, considered out of the perspective of "disc", the pivot word "wrap" tends to occur in a different dependency position. For example, the following sentence can be found in the English corpus:

> "Front **disc wraps**."

In English "wrap" tends to occur as a successor of "disc". A non-zero confusion coefficient allows us to account some degree of similarity to situations where the query (here "ディスク"(disc)) and the translation candidate (here "disc") tend to occur with the same pivot word (here "wrap"), but in different dependency positions.

6 Conclusions

Finding new translations of single words using comparable corpora is a promising method, for example, to assist the creation and extension of bilingual dictionaries. The basic idea is to first create context vectors of the query word, and all the candidate translations, and then, in the second step, to compare these context vectors. Previous work (Laroche and Langlais, 2010; Fung, 1998; Garera et al., 2009) suggests that for this task the cosine-similarity is a good choice to compare context vectors. For example, Garera et al. (2009) include the information of various context positions from the dependency-parse tree in one context vector, and, afterwards, compares these context vectors using the cosine-similarity. However, this makes the implicit assumption that all context positions are equally important, and, furthermore, that context from *different* context positions does not need to be compared with each other. To overcome these limitations, we suggested to use a generalization of the cosine similarity which performs a linear transformation of the context vectors, before applying the cosine similarity. The linear transformation can be described by a positive-definite matrix A. We defined the optimal matrix A by using a Bayesian probabilistic model. We demonstrated that it is feasible to approximate the optimal matrix A by using MCMC-methods.

Our experimental results suggest that it is beneficial to weight context positions individually. For example, we found that predecessor and successor should be stronger weighted than sibling, and sentence information. Whereas, the latter two are also important, having a total weight of around 40%. Furthermore, we showed that for languages as different as Japanese and English it can be helpful to compare also *different* context positions across both languages. The proposed method constantly outperformed the baseline method. Top 1 accuracy increased by up to 2% percent points and Top 20 by up to 4% percent points.

For future work, we consider to use different parameterizations of the matrix A which could lead to even higher improvement in accuracy. Furthermore, we consider to include, and weight additional features like transliteration similarity.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments. This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan). The first author is supported by the MEXT Scholarship and by an IBM PhD Scholarship Award.

References

D. Andrade, T. Nasukawa, and J. Tsujii. 2010. Robust measurement and comparison of context similarity for finding translation pairs. In *Proceedings of the International Conference on Computational Linguistics*, pages 19–27.

D. Andrade, T. Matsuzaki, and J. Tsujii. 2011. Effective use of dependency structure for bilingual lexicon creation. In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 80–92. Springer Verlag.

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.

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.

N. Garera, C. Callison-Burch, and D. Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 129–137. Association for Computational Linguistics.

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.

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.

F. Laws, L. Michelbacher, B. Dorow, C. Scheible, U. Heid, and H. Schütze. 2010. A linguistically grounded graph model for bilingual lexicon extraction. In *Proceedings of the International Conference on Computational Linguistics*, pages 614–622. International Committee on Computational Linguistics.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 91–98. Association for Computational Linguistics.

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.

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.

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.

T. Utsuro, T. Horiuchi, K. Hino, T. Hamamoto, and T. Nakayama. 2003. Effect of cross-language IR in bilingual lexicon acquisition from comparable corpora. In *Proceedings of the conference on European chapter of the Association for Computational Linguistics*, pages 355–362. Association for Computational Linguistics.

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. 2003. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, pages 521–528.