Dependency-based Sentence Alignment for Multiple Document Summarization

Tsutomu HIRAO and **Jun SUZUKI** and **Hideki ISOZAKI** and **Eisaku MAEDA**
NTT Communication Science Laboratories, NTT Corp.
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
{hirao,jun,isozaki,maeda}@cslab.kecl.ntt.co.jp

Abstract

In this paper, we describe a method of automatic sentence alignment for building *extracts* from *abstracts* in automatic summarization research. Our method is based on two steps. First, we introduce the “dependency tree path” (DTP). Next, we calculate the similarity between DTPs based on the ESK (Extended String Subsequence Kernel), which considers sequential patterns. By using these procedures, we can derive one-to-many or many-to-one correspondences among sentences. Experiments using different similarity measures show that DTP consistently improves the alignment accuracy and that ESK gives the best performance.

1 Introduction

Many researchers who study automatic summarization want to create systems that generate *abstracts* of documents rather than *extracts*. We can generate an *abstract* by utilizing various methods, such as sentence compaction, sentence combination, and paraphrasing. In order to implement and evaluate these techniques, we need large-scale corpora in which the original sentences are aligned with summary sentences. These corpora are useful for training and evaluating sentence extraction systems. However, it is costly to create these corpora.

Figure 1 shows an example of summary sentences and original sentences from TSC-2 (Text Summarization Challenge 2) multiple document summarization data (Okumura et al., 2003). From this example, we can see many-to-many correspondences. For instance, summary sentence (A) consists of a part of source sentence (A). Summary sentence (B) consists of parts of source sentences (A), (B), and (C). It is clear that the correspondence among the sentences is very complex. Therefore, robust and accurate alignment is essential.

Source(A): まず、韓国や台湾などの競合他社から価格攻勢が強まる可能性の高い**次世代メモリー６４メガビットＤＲＡＭ（記憶保持動作が必要な随時読み出し書き込みメモリー）への新規投資を中止**し、今後は**付加価値の高い**システム・オン・チップや次世代の２５６メガＤＲＡＭの**開発投資にシフトし**ていく。
First, **we stop the new investment of 64-Mega bit memory** from competitive companies, such as in Korea or Taiwan, and **we begin the investment for development of valuable system-on-chip or 256-Mega bit DRAM** from now on.

Source(B): また、**長期的には**半導体事業のうち**パソコン用などの汎（はん）用ＤＲＡＭ**事業の割合を減らしていく方針だ。
On a long-term target, we plan to reduce the rate of **general-purpose semiconductor enterprises that produce DRAM for personal computers**.

Source(C): 今後、必要な**ＤＲＡＭは台湾メーカーから供給する**方針だ。
From now on, **we will be supplied with DRAM from Taiwan**.

Summary(A): 次世代メモリー６４メガビットＤＲＡＭ（記憶保持動作が必要な随時読み出し書き込みメモリー）への新規投資を中止。
We stopped the new investment of 64-Mega bit DRAM.

Summary(B): 付加価値の高い開発投資にシフトし、長期的にはパソコン用などの汎用ＤＲＡＭは台湾メーカーから供給する。
We begin the investment for valuable development and will be supplied with general-purpose DRAMs for personal computers from Taiwan in the long run.

Figure 1: An example of summary sentences and their source sentences from TSC-2 multiple document summarization data. Underlined strings are used in summary sentences.

In order to achieve such alignment, we need not only syntactic information but also semantic information. Therefore, we combine two methods. First, we introduce the “dependency tree path” (DTP) for syntactic information. Second, we introduce the “Extended String Subsequence Kernel” (ESK) for semantic information.

Experimental results using different similarity measures show that DTP consistently improves alignment accuracy and ESK enhances the performance.

Sentence 1: 私が近所の警察に落し物を届けた.
watashi ga kinjo no keisatsu ni otosimono wo todoke ta.

Sentence 2: 近所の警察に落し物を私が届けた.
kinjo no keisatsu ni otoshimono wo watashi ga todoke ta.

Figure 2: Examples of sentences that have the same meaning.

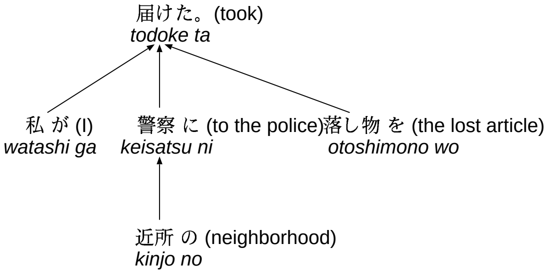

Figure 3: An example of a dependency tree.

2 Related Work

Several methods have been proposed to realize automatic alignment between *abstracts* and sentences in source documents.

Banko et al. (1999) proposed a method based on sentence similarity using bag-of-words (BOW) representation. For each sentence in the given *abstract*, the corresponding source sentence is determined by combing the similarity score and heuristic rules. However, it is known that bag-of-words representation is not optimal for short texts like single sentences (Suzuki et al., 2003).

Marcu (1999) regards a sentence as a set of “units” that correspond to clauses and defines similarity between units based on BOW representation. Next, the best source sentences are extracted in terms of “unit” similarity. Jing et al. (Jing and McKeown, 1999) proposed bigram-based similarity using the Hidden Markov Model. Barzilay (Barzilay and Elhadad, 2003) combines edit distance and context information around sentences. However, these three methods tend to be strongly influenced by word order. When the summary sentence and the source sentences disagree in terms of word order, the methods fail to work well.

The supervised learning-based method called SimFinder was proposed by Hatzivassiloglou et al. (Hatzivassiloglou et al., 1999; Hatzivassiloglou et al., 2001). They translate a sentence into a feature vector based on word counts and proper nouns, and so on, and then sentence pairs are classified into “similar” or not. Their approach is effective when a lot of training data is available. However, the human cost of making this training data cannot be disregarded.

3 An Alignment Method based on Syntax and Semantics

For example, Figure 2 shows two sentences that have different word order but the same meaning. The English translation is “I took the lost article to the neighborhood police.”

Since conventional techniques other than BOW are strongly influenced by word order, they are fragile when word order is damaged.

3.1 Dependency Tree Path (DTP)

When we unify two sentences, some elements become longer, and word order may be changed to improve readability. When we rephrase sentences, the dependency structure does not change in many cases, even if word order changes. For example, the two sentences in Figure 2 share the same dependence structure shown in Figure 3. Therefore, we transform a sentence into its dependency structure. This allows us to consider a sentence as a set of dependency tree paths from a leaf to the root node of the tree.

For instance, the two sentences in Figure 2 can be transformed into the following DTPs.

- 私 が 届け た (I took)
  watashi ga todoke ta
- 近所 の 警察 に 届け た (took to the neighborhood police)
  kinjo no keisatsu ni todoke ta
- 落し物 を 届け た (took the lost article)
  otoshimono wo todoke ta.

3.2 An Alignment Algorithm using DTPs

In this section, we describe a method that aligns source sentences with the summary sentences in an *abstract*.

Our algorithm is very simple. We take the corresponding sentence to be the one whose DTP is most similar to that of the summary sentence. The algorithm consists of the following steps:

Step 0 Transform all source sentences into DTPs.

Step 1 For each sentence “α” in the *abstract*, apply Step 2 and Step 3.

Step 2 Transform “α” into a DTP set. Here, $F(\alpha)$ denotes α’s DTP set. $F(s_i)$ denotes the DTP set of the i-th source sentences.

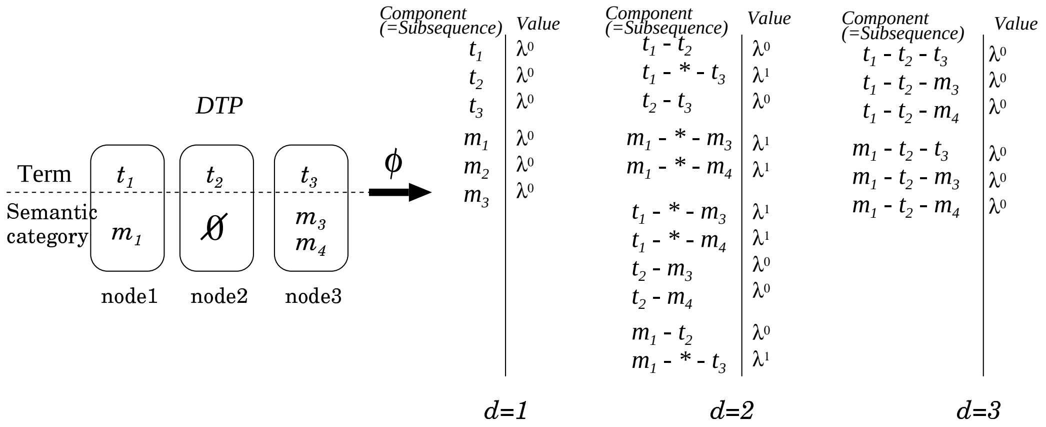

Figure 4: ESK with node sequence.

Step 3 For each $P_\alpha (\in F(\alpha))$, we align an optimal source sentence as follows:
We define sim$(P_\alpha, s_i) \overset{\text{def}}{=}$ max sim(P_α, P).
Here, $P \in F(s_i)$,
where, for P_α, we align a source sentence that satisfies $\text{argmax}_{s_i \in \text{source}} \, \text{sim}(P_\alpha, s_i)$.

The above procedure allows us to derive many-to-many correspondences.

3.3 Similarity Metrics

We need a similarity metric to rank DTP similarity. The following cosine measure (Hearst, 1997) is used in many NLP tasks.

$$\text{sim}_{\text{cos}}(o_1, o_2) = \frac{\sum_t w_{t,o_1} w_{t,o_2}}{\sqrt{\sum_t w_{t,o_1}^2 \sum_t w_{t,o_2}^2}} \qquad (1)$$

Here, w_{t,o_1}, w_{t,o_2}, denote the weight of term t in texts o_1, o_2, respectively. Note that syntactic and semantic information is lost in the BOW representation.

In order to solve this problem, we use similarity measures based on word co-occurrences. As an example of it s application, N-gram co-occurrence is used for evaluating machine translations (Papineni et al., 2002). String Subsequence Kernel (SSK) (Lodhi et al., 2002) and Word Sequence Kernel (WSK) (Cancedda et al., 2003) are extensions of n-gram-based measures used for text categorization.

In this paper, we compare WSK to its extension, the Extended String Subsequence Kernel (ESK). First, we describe WSK. WSK receives two sequences of words as input and maps each of them into a high-dimensional vector space. WSK's value is just the inner product of the two vectors.

Table 1: Components of vectors corresponding to 'abaca' and 'abbab.' Bold letters indicate common subsequences.

subsequence	abaca	abbab
abb	0	1+2λ^2
aba	1 + λ^2	2λ
abc	λ	0
aab	0	λ^2
aac	λ	0
aaa	λ^2	0
aca	λ^2+1	0
ab	1	2+λ + λ^3
aa	2λ+λ^3	λ^2
ac	1+λ^2	0
ba	1+λ^2	1+λ
bb	−	1+λ+λ^2
bc	λ	0
ca	1	0
a	3	2
b	1	3
c	1	0

For instance, the WSK value for 'abaca' and 'abbab' is determined as follows. A subsequence whose length is three or less is shown in Table 1. Here, λ is a decay parameter for the number of skipped words. For example, subsequence 'aba' appears in 'abaca' once without skips. In addition, it appears again with two skips, i.e., 'ab**a.' Therefore, abaca's vector has "1+λ^2" in the component corresponding to 'aba.' From Table 1, we can calculate the WSK value as follows:

$$\begin{aligned} K_{\text{wsk}}(\text{`}abaca', \text{`}abbab') &= \overbrace{(1+\lambda^2)\times 2\lambda}^{aba} \\ &+ \overbrace{1 \times (2+\lambda+\lambda^3)}^{ab} \end{aligned}$$

Table 2: Description of TSC data

	single	multiple
# of doc clusters	–	30
# of docs	30	224
# of sentences	881	2425
# of characters	34112	111472

$$
\begin{aligned}
&+ \overbrace{(2\lambda + \lambda^3) \times \lambda^2}^{aa} \\
&+ \overbrace{(1 + \lambda^2) \times (1 + \lambda)}^{ba} \\
&+ \overbrace{3 \times 2}^{a} + \overbrace{1 \times 3}^{b}. \qquad (2)
\end{aligned}
$$

In this way, we can measure the similarity between two texts. However, WSK disregards synonyms, hyponyms, and hypernyms. Therefore, we introduce ESK, an extension of WSK and a simplification of HDAG Kernel (Suzuki et al., 2003). ESK allows us to add word senses to each word. Here, we do not try to disambiguate word senses, but use all possible senses listed in a dictionary. Figure 4 shows an example of subsequences and their values. The use of word sense yields flexible matching even when paraphrasing is used for summary sentences.

Formally, ESK is defined as follows.

$$K_{\text{esk}}(T, U) = \sum_{m=1}^{d} \sum_{t_i \in T} \sum_{u_j \in U} K_m(t_i, u_j) \qquad (3)$$

$$K_m(t_i, u_j) = \begin{cases} val(t_i, u_j) & \text{if} \quad m = 1 \\ K'_{m-1}(t_i, u_j) \cdot val(t_i, u_j) & \end{cases} \qquad (4)$$

Here, $K'_m(t_i, u_j)$ is defined as follows. t_i and u_j are nodes of T and U, respectively. The function $val(s, t)$ returns the number of attributes common to given nodes s and t.

$$K'_m(t_i, u_j) = \begin{cases} 0 \quad \text{if} \quad j = 1 \\ \lambda K'_m(t_i, u_{j-1}) + K''_m(t_i, u_{j-1}) \end{cases} \qquad (5)$$

$K''_m(t_i, u_j)$ is defined as follows:

$$K''_m(t_i, u_j) = \begin{cases} 0 \quad \text{if} \quad i = 1 \\ \lambda K''_m(t_{i-1}, u_j) + K_m(t_{i-1}, u_j). \end{cases} \qquad (6)$$

Table 3: The distribution of aligned original sentences corresponding to one summary sentence.

# of org. sents.		1	2	3>
A1	Short	167 / (0.770)	49 / (0.226)	1 / (0.005)
	Long	283 / (0.773)	73 / (0.199)	10 / (0.027)
A2	Short	157 / (0.762)	46 / (0.223)	3 / (0.015)
	Long	299 / (0.817)	59 / (0.161)	11 / (0.022)
A3	Short	198 / (0.846)	34 / (0.145)	2 / (0.009)
	Long	359 / (0.890)	39 / (0.097)	5 / (0.012)
B1	Short	295 / (0.833)	45 / (0.127)	14 / (0.040)
	Long	530 / (0.869)	65 / (0.107)	15 / (0.025)
B2	Short	156 / (0.667)	58 / (0.248)	20 / (0.085)
	Long	312 / (0.698)	104 / (0.233)	31 / (0.069)
B3	Short	191 / (0.705)	62 / (0.229)	18 / (0.066)
	Long	392 / (0.797)	76 / (0.154)	24 / (0.048)

Table 4: The distribution of aligned summary sentences corresponding to one original sentence.

# of sum. sents.		1	2	3 >
A1	Short	268 / (1.000)	0	0
	Long	458 / (0.994)	2 / (0.006)	0
A2	Short	258 / (1.000)	0	0
	Long	440 / (1.000)	0	0
A3	Short	272 / (1.000)	0	0
	Long	450 / (1.000)	0	0
B1	Short	406 / (0.974)	11 / (0.026)	0
	Long	660 / (0.964)	22 / (0.032)	2 / (0.004)
B2	Short	317 / (0.975)	8 / (0.025)	0
	Long	550 / (0.945)	31 / (0.053)	1 / (0.002)
B3	Short	364 / (0.989)	4 / (0.011)	0
	Long	583 / (0.965)	16 / (0.025)	5 / (0.010)

Finally, we define the similarity measure by normalizing ESK. This similarity can be regarded as an extension of the cosine measure.

$$\text{sim}_{\text{esk}}(T, U) = \frac{K_{\text{esk}}(T, U)}{\sqrt{K_{\text{esk}}(T, T) K_{\text{esk}}(U, U)}}. \qquad (7)$$

4 Evaluation Settings

4.1 Corpus

We used the TSC2 corpus which includes both single and multiple document summarization data. Table 2 shows its statistics. For each data set, each of three experts made short abstracts and long abstracts.

For each data, summary sentences were aligned with source sentences. Table 3 shows the distribution of the numbers of aligned original sentences for each summary sentence. The values in brackets are percentages. Table 4 shows the distribution of the number of aligned summary sentences for each original sentence. These tables show that sentences are often split and reconstructed. In particular, multiple document summarization data exhibit

Table 5: Evaluation results w/o DTP (single documents).

		ESK	WSK	BOW	2-gram	3-gram	TREE
A1	Short	0.951	**0.958**	0.906	0.952	0.948	0.386
	Long	0.951	0.959	0.916	**0.961**	0.959	0.418
A2	Short	0.938	**0.954**	0.916	0.945	0.950	0.322
	Long	0.968	**0.973**	0.940	0.966	0.972	0.476
A3	Short	0.927	**0.951**	0.875	0.926	0.926	0.436
	Long	**0.967**	0.966	0.926	0.961	0.962	0.547

Table 6: Evaluation results with DTP (single documents).

		DTP(ESK)	DTP(WSK)	DTP(BOW)	DTP(2-gram)	DTP(3-gram)
A1	Short	**0.966 (2,1.00)**	0.957 (2,0.10)	0.955	0.952	0.952
	Long	0.960 (4,0.20)	0.957 (2,0.20)	0.960	0.951	0.949
A2	Short	**0.973 (3,0.60)**	0.957 (2,0.10)	0.959	0.957	0.956
	Long	**0.977 (4,0.20)**	0.974 (2,0.95)	0.972	0.973	0.975
A3	Short	0.962 (3,0.70)	0.962 (3,0.50)	**0.964**	0.962	0.960
	Long	0.967 (3,0.70)	**0.969 (2,0.20)**	0.962	0.960	0.960

Table 7: Effectiveness of DTP (single documents).

		ESK	WSK	BOW	2-gram	3-gram
A1	Short	+1.5	−0.1	+4.9	±0	+0.8
	Long	+0.9	−0.2	+4.4	0.1	−1.0
A2	Short	+3.5	+0.3	+4.3	+1.2	+0.6
	Long	+0.9	+0.1	+3.2	+0.7	+0.3
A3	Short	+3.5	+1.1	+8.9	+3.6	+3.4
	Long	±0	+0.3	+3.6	−0.1	−0.2

very complex correspondence because various summarization techniques such as sentence compaction, sentence combination, and sentence integration are used.

4.2 Comparison of Alignment Methods

We compared the proposed methods with a baseline algorithm using various similarity measures.

Baseline

This is a simple algorithm that compares sentences to sentences. Each summary sentence is compared with all source sentences and the top n sentences that have a similarity score over a certain threshold value τ are aligned.

DTP-based Method

This method was described in Section 3.2. In order to obtain DTPs, we used the Japanese morphological analyzer ChaSen and the dependency structure analyzer CaboCha (Kudo and Matsumoto, 2002).

4.2.1 Similarity Measures

We utilized the following similarity measures.

BOW BOW is defined by equation (1). Here, we use only nouns and verbs.

N-gram This is a simple extension of BOW. We add n-gram sequences to BOW. We examined "2-gram" (unigram + bigram) and "3-gram,"(unigram + bigram + trigram).

TREE The Tree Kernel (Collins and Duffy, 2001) is a similarity measure based on the number of common subtrees. We regard a sentence as a dependency structure tree.

WSK We examined $d = 2$, 3, and 4, and $\lambda = 0.05, 0.1, 0.15 \cdots, 1$.

ESK We used the Japanese lexicon Goi-Taikei (Ikehara et al., 1997), to obtain word senses. The parameters, d and λ, were changed on the same Conditions as above.

4.3 Evaluation Metric

Each system's alignment output was scored by the average F-measure. For each summary sentence, the following F-measure was calculated.

$$\text{F-measure} = \frac{1+\gamma^2}{(\frac{1}{\text{Precision}} + \gamma^2 \frac{1}{\text{Recall}})} \qquad (8)$$

Here, Precision $= b/a$ and Recall $= b/c$, where a is the number of source sentences aligned by a system for the summary sentence. b is the number of correct source sentences in the output. c is the number of source sentences aligned by the human expert. We set γ to 1, so this F-measure was averaged over all summary sentences.

5 Results and Discussion

5.1 Single Document Summarization Data

Table 5 shows the results of the baseline method (*i.e.,* without DTPs) with the best τ; Table 6 shows

Table 8: Evaluation results w/o DTP (multiple documents).

		ESK	WSK	BOW	2-gram	3-gram	TREE
B1	Short	0.609	0.547	0.576	**0.644**	0.638	0.127
	Long	0.674	0.627	0.655	**0.714**	0.711	0.223
B2	Short	0.622	0.660	0.590	0.668	**0.680**	0.161
	Long	0.742	0.769	0.690	0.751	**0.761**	0.236
B3	Short	0.683	0.712	0.654	**0.733**	0.729	0.158
	Long	0.793	**0.821**	0.768	0.805	0.817	0.280

Table 9: Evaluation results with DTP (multiple documents).

		DTP(ESK)	DTP(WSK)	DTP(BOW)	DTP(2-gram)	DTP(3-gram)
B1	Short	**0.746** (2,0.85)	0.734 (2,0.55)	0.719	0.725	0.728
	Long	**0.802** (3,0.85)	0.797 (2,0.65)	0.784	0.797	0.797
B2	Short	0.726 (2,0.65)	**0.741** (3,0.25)	0.710	0.720	0.721
	Long	**0.808** (2,0.55)	0.800 (3,0.05)	0.797	0.797	0.794
B3	Short	**0.790** (2,0.55)	0.786 (3,0.05)	0.748	0.768	0.760
	Long	0.845 (3,0.60)	**0.861** (2,0.40)	0.828	0.835	0.830

Table 10: Effectiveness of DTP (multiple documents).

		ESK	WSK	BOW	2-gram	3-gram
B1	Short	+13.7	+18.7	+14.3	+8.1	+9.6
	Long	+12.8	+17.0	+12.9	+8.3	+8.6
B2	Short	+10.4	+8.1	+12.0	+5.2	+4.1
	Long	+5.6	+3.1	+10.7	+4.6	+3.3
B3	Short	+10.7	+7.4	+9.4	+3.5	+3.1
	Long	+5.2	+4.0	+6.0	+3.0	+1.3

the results of using DTPs with the best d and λ, which are shown in brackets. From the results, we can see the effectiveness of DTPs because Table 6 shows better performance than Table 5 in most cases. Table 7 shows the difference between Tables 5 and 6. DTPs improved the results of BOW by about five points. The best result is DTP with ESK. However, we have to admit that the improvements are relatively small for single document data. On the other hand Tree Kernel did not work well since it is too sensitive to slight differences. This is known as a weak point of Tree Kernel (Suzuki et al., 2003).

According to the tables, BOW is outperformed by the other methods except Tree Kernel. These results show that word co-occurrence is important. Moreover, we see that sequential patterns are better than consequential patterns, such as the N-gram.

Without DPTs, ESK is worse than WSK. However, ESK becomes better than WSK when we use DTPs. This result implies that word senses are disambiguated by syntactic information, but more examination is needed.

5.2 Multiple Document Summarization Data

Table 8 shows the results of the baseline method with the best τ for multiple document data while Table 9 shows the result of using DTPs with the best d and λ, (in brackets). Compared with the single document summarization results, the F-measures are low. This means that the sentence alignment task is more difficult in multiple document summarization than in single document summarization. This is because sentence compaction, combination, and integration are common.

Although the results show the same tendency as the single document summarization case, more improvements are noticed. Table 10 shows the difference between Tables 8 and 9. We see improvements in 10 points in ESK, WSK, and BOW. In multiple document summarization, sentences are often reorganized. Therefore, it is more effective to decompose a sentence into DTP sets and to compute similarity between the DTPs.

Moreover, DTP(ESK) is once again superior to DTP(WSK).

5.3 Parameter Tuning

For ESK and WSK, we have to choose parameters, d and λ. However, we do not know an easy way of finding the best combination of d and λ. Therefore, we tuned these parameters for a development set. The experimental results show that the best d is 2 or 3. However, we could not find a consistently optimal value of λ. Figure 5 shows the F-measure with various λ for $d = 2$. The results shows that the F-measure does not change very much in the middle range λ, [0.4,0.6] which suggests that good results are possible by using a middle range λ.

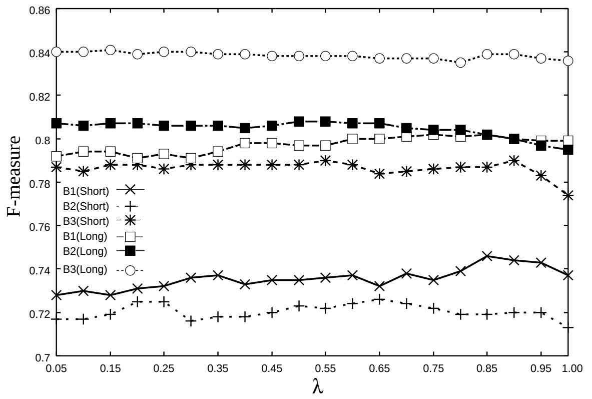

Figure 5: F-measures with various λ values($d = 2$).

6 Conclusion

This paper introduced an automatic sentence alignment method that integrates syntax and semantic information. Our method transforms a sentence into a DTP set and calculates the similarity between the DTPs by using ESK. Experiments on the TSC (Text Summarization Challenge) corpus, which has complex correspondence, showed that the introduction of DTP consistently improves alignment accuracy and that ESK gave the best results.

References

M. Banko, V. Mittal, M. Kantrowitz, and J. Goldstein. 1999. Generating Extraction-Based Summaries from Hand-Written Summaries by Aligning Text Spans. *Proc. of the 4th Conference of the Pacific Association for Computational Linguistics*.

R. Barzilay and N. Elhadad. 2003. Sentence Alignment for Monolingual Comparable Corpora. *Proc. of the Empirical Methods for Natural Language Processing 2003*, pages 25–32.

N. Cancedda, E. Gaussier, C. Goutte, and J-M. Renders. 2003. Word-Sequence Kernels. *Journal of Machine Learning Research*, 3(Feb):1059–1082.

M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proc. of Neural Information Processing Systems (NIPS'2001)*.

V. Hatzivassiloglou, J.L. Klavans, and E. Eskin. 1999. Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. *Proc. of the Empirical Methods for Natural Language Processing 1999*, pages 203–212.

V. Hatzivassiloglou, J.L. Klavans, M.L. Holcombe, R. Barzilay, M-Y. Kan, and K. R. McKeown. 2001. SimFinder: A Flexible Clustering Tool for Summarization. *Proc. of the Workshop on Automatic Summarization 2001*, pages 41–49.

M-A. Hearst. 1997. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, 23(1):33–64.

S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. 1997. *Goi-Taikei – A Japanese Lexicon (in Japanese)*. Iwanami Shoten.

H. Jing and K. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. *Proc. of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 129–136.

T. Kudo and Y. Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. *Proc. of the 6th Conference on Natural Language Learning*, pages 63–69.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text Classification using String Kernel. *Journal of Machine Learning Research*, 2(Feb):419–444.

D. Marcu. 1999. The Automatic Construction of Large-scale Corpora for Summarization Research. *Proc. of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 137–144.

M. Okumura, T. Fukusima, and H. Nanba. 2003. Text Summarization Challenge 2 - Text Summarization Evaluation at NTCIR Workshop 3. *HLT-NAACL 2003 Workshop: Text Summarization (DUC03)*, pages 49–56.

S. Papineni, S. Roukos, T. Ward, and W-J Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 62–66.

J. Suzuki, T. Hirao, Y. Sasaki, and E. Maeda. 2003. Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data. *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 32–39.