テキスト平易化コーパス構築のための文の分割を考慮した文間類似度計算法

永井 優城　　能地 宏　　松本 裕治
奈良先端科学技術大学院大学 情報学研究科
{nagai.yuki.nr9, noji, matsu}@is.naist.jp

1 はじめに

テキスト平易化とは難しい文を同じ意味の平易な文に変換するタスクである. テキスト平易化は子供や第二言語学習者, 難読症患者といった人々の読解支援に役立つだけでなく, 他の自然言語処理タスクの前処理としても有用である.

近年, 英語では統計的機械翻訳を用いてテキスト平易化を実現する研究が多く行われている [7, 2]. 学習に用いるコーパスは人手で構築するとコストが高いため, English Wikipedia と Simple English Wikipedia から機械的に文アライメントを取り, パラレルコーパスを構築することにより作成されている.

単言語パラレルコーパスの構築は文間類似度の計算, 文アライメントのアルゴリズムという 2 つの要素により精度が左右される. テキスト平易化は語彙の平易化, 長い文の分割, 単語の並び替え, 難しい単語の削除といった 4 つの操作により実現される事が多いが, これらのパラレルコーパスを作成する際, 長い文の分割を考慮してコーパスを作成している研究は少ない. Hwang らは文間類似度の計算の際に, English Wikipedia 側の文を構文解析し, その部分木と Simple English 側の文の文間類似度を計算し文アライメントを取った [3]. しかし, Hwang らの手法は 1 対 1 の文アライメントしか許容しておらず, 本来文分割が行われた文は 1 対多の文アライメントがなされるべきである.

そこで本稿では文間類似度の計算の際に文分割を考慮し, 1 対多のアライメントを許容する文アライメントアルゴリズムを適用することにより精度の高い文アライメントの作成を目指す. 人手でアノテーションされた文アライメントのテストデータを用いて実験したところ, 先行研究よりも高精度で文アライメントを作成できることがわかった.

2 関連研究

Zhu ら [7] は English Wikipedia と Simple English Wikipedia から tf-idf を用いて文ベクトルを作り, コサイン類似度を用いてパラレルコーパス[1]を構築した. Zhu らの手法は 2 つの文の文間類似度がある閾値を超えるかどうかでアライメントを決定しており, 1 体多のアライメントは取れているが文分割を明示的に考慮してはいない.

Coster and Kauchack[2] は Zhu らと同様に tf-idf を用いて文ベクトルを作り, Barzilay らの文アライメントアルゴリズム [1] を適用し文の順番, 文の分割を考慮し文アライメントをつけ, テキスト平易化のコーパスを作成した[2]. Coster and Kauchack の文アライメントアルゴリズムは文分割を考慮し, 1 対多のアライメントが可能となっているが, 文間類似度の計算時には分割を考慮しておらず, それぞれ分割された文を独立に計算したスコアを元に文アライメントをつけている.

Hwang ら [3] は人手により English Wikipedia と Simple English Wikipedia の文アライメントを人手でアノテーションした データセットを作成し, 過去の手法と定量的な比較を行った. また Hwang らは WikNet を用いて類義語を考慮しながら文間類似度を計算した. また English Wikipedia の文を構文解析し, 部分木を用いることにより長い文の分割を考慮し, 文アライメントアルゴリズムとして Greedy アルゴリズムを提案しコーパスを作成した[3]. 一方で文アライメントは 1 対 1 に制限されており, 長い文が 2 つの文に分割される場合を考慮していない.

Kajiwara and Komachi[4] は Song ら [6] の単語分散表現を用いた文間類似度計算手法をテキスト平易化コーパスの構築に適用し, 類義語を考慮しながら文間

[1] https://www.ukp.tu-darmstadt.de/data/sentence-simplification/simple-complex-sentence-pairs/
[2] http://www.cs.pomona.edu/ dkauchak/simplification/
[3] http://ssli.ee.washington.edu/tial/projects/simplification/

類似度の計算を行い, コーパスを作成した[4]. アライメント手法は Zhu らと同様に文間類似度がある閾値を超えるかどうかでアライメントを決定しており, 文分割を明示的に考慮していない.

3 手法

本節では文分割を考慮した文間類似度計算手法について述べ, 次に文アライメントアルゴリズム, 文間類似度のランキングについて述べる. 提案手法の文アライメントアルゴリズムは Hwang らの Greedy アルゴリズムを文分割を考慮できるように拡張したものである.

3.1 文分割を考慮した類似度計算

記事単位で対応が取れた English Wikipedia と Simple English Wikipedia の記事を入力とする. English Wikipedia の文を N_i, Simple English Wikipedia の文を S_j とし, 文 X と文 Y の文間類似度計算関数を $Sim(X,Y)$ とする. N_i と S_j の文間類似度を w_{ij} とし, 全ての文対の組み合わせの文間類似度を要素にもつ行列を W とする.

English Wikipedia の長い文が分割される際, 分割される文には when や and などの接続詞, which, who 等の関係代名詞等による節による分割が多い. また Simple English Wikipedia 上で分割された文は連接している可能性が高い. そこで N_i と S_j の文間類似度を計算する際, S_j とその連接する S_{j-1}, S_{j+1} とそれぞれ結合した $(S_{j-1}+S_j)$, (S_j+S_{j+1}) との文間類似度を計算する. その中で最も N_i と文間類似度が高いものを w_{ij} の重みとする.

3.2 Append greedy アルゴリズム

重み行列 W の中で最も文間類似度重みが大きい文対 (N_{i^*}, S_{j^*}) を選びアライメントをつける. ここで $(i^*, j^*) = argmax_{(i,j)}(w_{ij})$ である. この際, S_{i^*} が連結された文からなる場合は N_{j^*} と連結される前の 2 つの文にアライメントをつける. その後 N_{j^*},S_{i^*} と関連する全ての文対を排除するため $\forall j, w_{i^*j} \leftarrow 0$ と $\forall i, w_{ij^*} \leftarrow 0$ とし, 重み行列の全ての重みが 0 になるまでこれを繰り返し, 文アライメントを作成する.

[4]https://github.com/tmu-nlp/sscorpus

3.3 文間類似度のランキング

Greedy アルゴリズムによって得られた文アライメントにおいて, N_i と S_j にアライメントがある場合, この重みを $Sim(N_i, S_j)$ で計算し, 全ての文をランキングする.

4 実験

4.1 実験設定

Hwang ら [3] のデータセット[5]を用いて実験を行った. このデータセットは English Wikipedia と Simple English Wikipedia の同一の記事の全ての文の組み合わせについて 4 段階の類似度で人手でラベル付している. 4 段階のラベルは Good, Good partial, Partial, Bad であり, Good : 2 つの文の意味が小さな省略等を除き, 完全に一致している, Good Partial : 片方の文がもう片方の文の意味を内包しているが, 片方にはない句や節が存在している. Partial : 2 つの文の意味は異なっているが, 似たような意味の句を含んでいる, Bad : 2 つの文が無関係である, といった関係をそれぞれ表す.

データセットは 46 記事, 全 67,853 文対 (277 Good, 281 Good partial, 117 Partial, 67,178 Bad) からなる. 過去の研究と同様に, 以下の 2 つの設定で実験を行う. 実験 1 では Good ラベルを正例, その他のラベルを負例としての 2 値分類を行う (Good vs. Others) . 実験 2 では Good ラベルと Good partial ラベルを正例, その他のラベルを負例としての 2 値分類を行う (Good and Good Partial vs. Others) .

評価は Precision-recall curve 上の Maximum F1 score (MaxF1) と The area under the curve (AUC) で行った. 文間類似度の計算の際には sentence-level の tf-idf の文ベクトルのコサイン類似度を用いた [5]. 3 節の提案手法を tf-idf append greedy とし, 比較手法として, 文分割を考慮せずに tf-idf のコサイン類似度と Hwang らの Greedy アルゴリズムを適用した手法 (tf-idf greedy), 過去の先行研究との比較を行った[6].

[5]http://ssli.ee.washington.edu/tial/projects/simplification/

[6]tf-idf append greedy, tf-idf greedy では good と good partial を明示的に区別はせず, アライメントがある, なしのラベル付を行った. Good と Good partial のラベル付を行う実験も行ったが, 明示的に区別しない実験よりも精度が低かった.

Method	MaxF1	AUC
Zhu et al. (2010)	0.550	0.509
Coster and Kauchak (2011)	0.564	0.495
Hwang et al. (2015)	0.712	0.694
Kajiwara and Komachi (2016)	**0.724**	**0.738**
tf-idf greedy	0.723	0.699
tf-idf append greedy	0.714	0.716

表 1: MaxF1, AUC for Good vs. Others.

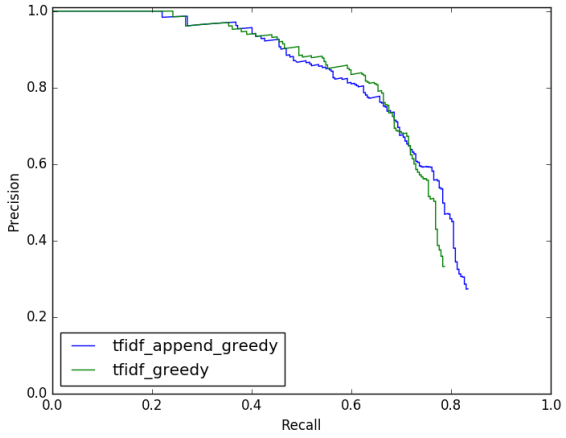

図 1: MaxF1, AUC for Good and Good Partial vs. Others.

4.2 結果

実験 1 の結果を表 1, Precision-recall-curve を図 1 に示す. Zhu らと Coster and Kauchack らの手法では文間類似度計算において tf-idf を用いており, 我々の文間類似度計算手法及び, アライメント手法が優れていることが確認できる. また Hwang ら, Kajiwara and Komachi の手法と比較すると tf-idf を用いるだけで類義語を考慮する文間類似度計算と同程度の精度が出せることがわかった. tf-idf greedy と tf-idf append greedy を比較すると, MaxF1 が 1 ポイント減少し, AUC が 1 ポイント向上した. Precision-recall curve のグラフでは Precision の減少, Recall の上昇が確認できる.

実験 2 の結果を表 1, Precision-recall-curve を図 2 に示す. 我々の手法が MaxF1, AUC ともに最も精度が高くなっている. tf-idf greedy と tf-idf append greedy を比較すると, MaxF1 が 3.8 ポイント上昇し, AUC が 7.5 ポイント上昇した. Precision-Recall Curve のグラフでは実験 1 とは異なり Precision の減少は見られず, Recall の上昇が確認できる. 先行研究での最高精度は Kajiwara and Komachi の手法であるが, これはハイパーパラメータを調整する必要とする手法である一方,

Method	MaxF1	AUC
Zhu et al. (2010)	0.431	0.391
Coster and Kauchak (2011)	0.564	0.495
Hwang et al. (2015)	0.607	0.529
Kajiwara and Komachi (2016)	0.638	0.618
tf-idf greedy	0.642	0.546
tf-idf apeend greedy	**0.680**	**0.621**

表 2: MaxF1, AUC for Good and Good Partial vs. Others.

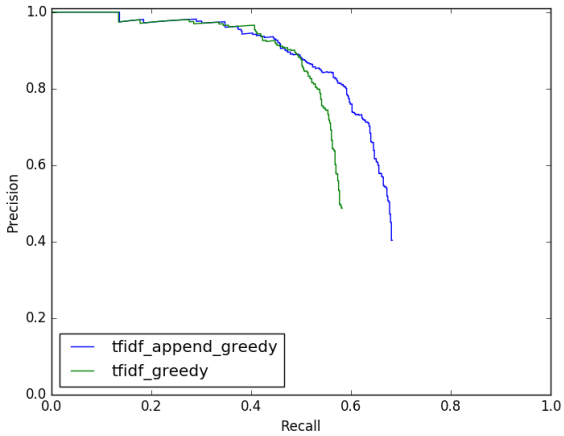

図 2: MaxF1, AUC for Good and Good Partial vs. Others.

提案手法はハイパーパラメータなしでこれを上回る精度となっている.

表 3 に文分割を考慮したアライメントにより, 長い文の分割を復元できた例を示す. Normal の行は English Wikipedia の文を表しており, 下の 2 行はその文に対して tf-idf greedy がアライメントした Simple English Wikipedia の文, tf-idf append greedy がアライメントした Simple Engilsh Wikipedia の文を表している. tf-idf greedy がアライメントした文の正解ラベルは Bad であり, tf-idf append greedy がアライメントした文の正解ラベルは両者共に Good Partial であり, 分割された文を復元できていることが確認できる.

5 結論

本稿では文の分割を考慮した文間類似度計算と 1 対多の文アライメントアルゴリズムについて述べた. 人手でラベル付されたデータセットによる評価実験により, 精度の向上を達成し, 文間類似度計算の際に文の分割を考慮することが有効であること確認できた.

Normal	They have one son and five daughters : Janet Rachel Huxley (born 20 April 1948) Stewart Leonard Huxley (born 19 December 1949) Camilla Rosalind Huxley (born 12 March 1952) Eleanor Bruce Huxley (born 21 February 1959) Henrietta Catherine Huxley (born 25 December 1960) Clare Marjory Pease Huxley (born 4 November 1962) He won the 1963 Nobel Prize in Physiology or Medicine for his experimental and mathematical work with Alan Hodgkin on the basis of nerve action potentials , the electrical impulses that enable the activity of an organism to be coordinated by a central nervous system .
tf-idf append greedy	Huxley won the 1963 Nobel Prize in Physiology or Medicine for his experimental and mathematical work with Alan Hodgkin on the basis of nerve action potentials .
	These are the electrical impulses that make nerve fibers work , and so the whole central nervous system .
tf-idf greedy	Huxley was the youngest son of the writer and editor Leonard Huxley by his second wife Rosalind Bruce .

表 3:

参考文献

[1] Regina Barzilay and Noemie Elhadad. *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, chapter Sentence Alignment for Monolingual Comparable Corpora. 2003.

[2] William Coster and David Kauchak. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 665–669, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[3] William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. Aligning sentences from standard wikipedia to simple wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 211–217, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[4] Tomoyuki Kajiwara and Mamoru Komachi. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1147–1158, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[5] Rani Nelken and Stuart M. Shieber. Towards robust context-sensitive sentence alignment for monolingual corpora. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[6] Yangqiu Song and Dan Roth. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1275–1280. Association for Computational Linguistics, 2015.

[7] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1353–1361, Beijing, China, August 2010. Coling 2010 Organizing Committee.