Neobility at SemEval-2017 Task 1: An Attention-based Sentence Similarity Model

WenLi Zhuang *
Shan-Si Elementary School
ChangHua County, Taiwan
bibo9901@gmail.com

Ernie Chang
Department of Linguistics
University of Washington
Seattle, WA 98195, USA
cyc025@uw.edu

Abstract

This paper describes a neural-network model which performed competitively (top 6) at the SemEval 2017 cross-lingual Semantic Textual Similarity (STS) task. Our system employs an attention-based recurrent neural network model that optimizes the sentence similarity. In this paper, we describe our participation in the multilingual STS task which measures similarity across English, Spanish, and Arabic.

1 Introduction

Semantic textual similarity (STS) measures the degree of equivalence between the meanings of two text sequences (Agirre et al., 2016). The similarity of the text pair can be represented as discrete or continuous values ranging from irrelevance (1) to exact semantic equivalence (5). It is widely applicable to many NLP tasks including summarization (Wong et al., 2008; Nenkova et al., 2011), generation and question answering (Vo et al., 2015), paraphrase detection (Fernando and Stevenson, 2008), and machine translation (Corley and Mihalcea, 2005).

In this paper, we describe a system that is able to learn context-sensitive features within the sentences. Further, we encode the sequential information with Recurrent Neural Network (RNN) and perform attention mechanism (Bahdanau et al., 2015) on RNN outputs for both sentences. Attention mechanism was performed to increase sensitivity of the system to words of similarity significance. We also optimize directly on the Pearson correlation score as part of our neural-based approach. Moreover, we include a pair feature adapter module that could be used to include more features to further improve performance. However, for this competition we include merely the TakeLab features (Šarić et al., 2012). [1]

2 Related Works

Most proposed approaches in the past adopted a hybrid of varying text unit sizes ranging from character-based, token-based, to knowledge-based similarity measure (Gomaa and Fahmy, 2013). The linguistic depths of these measures often vary between lexical, syntactic, and semantic levels.

Most solutions include an ensemble of modules that employs features coming from different unit sizes and depths. More recent approaches generally include the word embedding-based similarity (Liebeck et al., 2016; Brychcín and Svoboda, 2016) as a component of the final ensemble. The top performing team in 2016 (Rychalska et al., 2016) uses an ensemble of multiple modules including recursive autoencoders with WordNet and monolingual aligner (Sultan et al., 2016). UMD-TTIC-UW (He et al., 2016) proposes the MPCNN model that requires no feature engineering and managed to perform competitively at the 6th place. MPCNN is able to extract the hidden information using the Convolutional Neural Network (CNN) and adds an attention layer to extract the vital similarity information.

3 Methods

3.1 Model

Given two sentences $I_1 = \{w_1^1, w_2^1, ..., w_{n^1}^1\}$ and $I_2 = \{w_1^2, w_2^2, ..., w_{n^2}^2\}$, where w_{ij} denote the jth token of the ith sentence, embedded using a function ϕ that maps each token to a D-dimension trainable vector. Two sentences are encoded with

*The author is currently serving in his Alternative Military Service of Education.

[1] Our system and data is available at github.com/iamalbert/semval2017task1.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 164–169,
Vancouver, Canada, August 3 - 4, 2017.

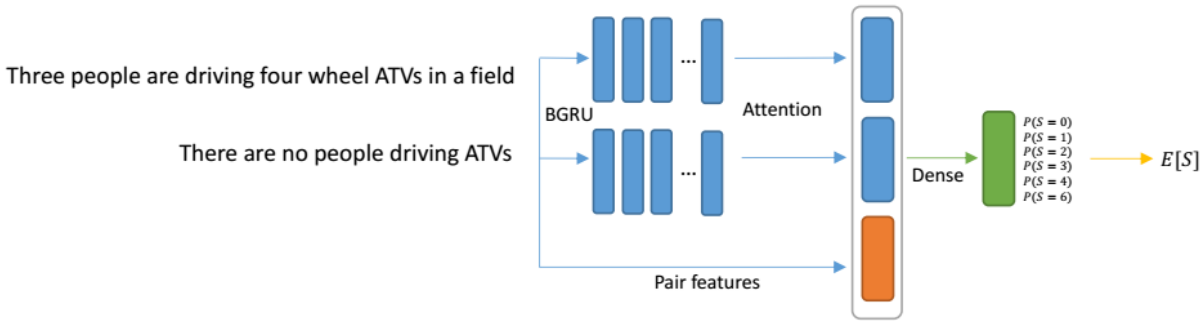

Figure 1: Illustration of model architecture

an attentitve RNN to obtain sentence embeddings u^1, u^2, respectively.

Sentence Encoder For each sentence, the RNN firstly converts w_j^i into $x_j^i \in R^{2H}$, using an bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) [2] by sequentially feeding w_j^i into the unit, in both forward and backward directions. The superscripts of w, x, a, u, n are omitted for clear notation.

$$\begin{aligned} x_i &= [x_i^F; x_i^B] \\ x_i^F &= \text{GRU}(x_{i-1}^F, w_i) \\ x_i^B &= \text{GRU}(x_{i+1}^B, w_i) \end{aligned} \quad (1)$$

Then, we attend each word x_j for different salience a_j and blend the memories $x_{1;n}$ into sentence embedding u:

$$\begin{aligned} a_j &\propto \exp(r^T \tanh(W x_i)) \\ u &= \sum_{j=1}^{n} a_j x_j \end{aligned} \quad (2)$$

where $W \in R^{2H \times 2H}$ and $r \in R^{2H}$ are trainable parameters.

Surface Features Inspired by the *simple* system described in (Šarić et al., 2012), We also extract surface features from the sentence pair as following:

- **Ngram Overlap Similarity:** These are features drawn from external knowledge like WordNet (Miller, 1995) and Wikipedia. We use both PathLen similarity (Leacock and Chodorow, 1998) and Lin similarity (Lin et al., 1998) to compute similarity between pairs of words w_i^1 and w_j^2 in I_1 and I_2, respectively. We employ the suggested preprocessing step (Šarić et al., 2012), and add both WordNet and corpus-based information to ngram overlap scores, which is obtained with the harmonic mean of the degree of overlap between the sentences.

- **Semantic Sentence Similarity:** We also compute token-based alignment overlap and vector space sentence similarity (Šarić et al., 2012). Semantic alignment similarity is computed greedily between all pairs of tokens using both the knowledge-based and corpus-based similarity. Scores are further enhanced with the aligned pair information. We obtain the weighted form of latent semantic analysis vectors (Turney and Pantel, 2010) for each word w, before computing the cosine similarity. As such, sentence similarity scores are enhanced with corpus-based information for tokens. The features are concatenated into a vector, denoted as m.

Scoring Let S be a descrete random variable over $\{0, 1, ..., 4, 5\}$ describing the similarity of the given sentence pair $\{I_1, I_2\}$. The representation of the given pair is the concatenation of u^1, u^2, and m, which is fed into an MLP with one hidden layer to calculate the estimated distribution of S.

$$\begin{aligned} p &= \begin{bmatrix} P(S=0) \\ P(S=1) \\ \vdots \\ P(S=5) \end{bmatrix} \\ &= \text{softmax}(V \tanh(U \begin{bmatrix} u^1 \\ u^2 \\ m \end{bmatrix})) \end{aligned} \quad (3)$$

Therefore, the score y is the expected value of

[2] We also explored Longer Short-Term Memory (LSTM), but find it more overfitting than GRU.

S:

$$y = E[S] = \sum_{i=0}^{5} iP(S=i) = v^T p \quad (4)$$

, where $v = [0, 1, 2, 3, 4, 5]^T$. The entire system is shown in Figure 1.

3.2 Word Embedding

We explore initializing word embeddings randomly or with pre-trained word2vec (Mikolov et al., 2013) of dimension 50, 100, 300, respectively. We found that the system works the best with 300-dimension word2vec embeddings.

3.3 Optimization

Let p^n, y^n be the predicted probability density and expected score and $\hat{y}^n$ be the annotated gold score of the n-th sample. Most of the previous learning-based models are trained to minimize the following objectives on a batch of N samples:

- Negative Log-likelihood (NLL) of p and $\hat{p}$ (Aker et al., 2016). The task is viewed as a classification problem for 6 classes.

$$L_{\text{NLL}} = \sum_{n=1}^{N} -\log p^n_{t^n}$$

, where t^n is $\hat{y}^n$ rounded to the nearest integer.

- Mean square error (MSE) between y^n and $\hat{y}^n$ (Brychcín and Svoboda, 2016).

$$L_{\text{MSE}} = \frac{1}{N}\sum_{n=1}^{N}(y^n - \hat{y}^n)^2$$

- Kullback-Leibler divergence (KLD) of p^n and gold distribution $\hat{p}^n$ estimated by $\hat{y}^n$:

$$L_{\text{KLD}} = \sum_{n=1}^{N}\left(\sum_{i=1}^{6}\hat{p}^n_i \log\frac{\hat{p}^n_i}{p^n_i}\right)$$

where

$$\hat{p}^n_i = \begin{cases} \hat{y}^n - \lfloor\hat{y}^n\rfloor, & \text{if } i = \lfloor\hat{y}^n\rfloor + 1 \\ \lfloor\hat{y}^n\rfloor + 1 - \hat{y}^n, & \text{if } i = \lfloor\hat{y}^n\rfloor \\ 0, & \text{otherwise} \end{cases}$$

(Li and Huang, 2016; Tai et al., 2015). For each n, there exists some k such that $\hat{p}^n_k = 1$ and $\forall h \neq k, \hat{p}^n_h = 0$, KLD is identical to NLL.

However, the evaluation metric of this task is Pearson Correlation Coefficient (PCC), which is invariant to changes in location and scale of y^n but none of the above objectives can reflect it. Here we use an example to illustrate that MSE and KLD can even report an inverse tendency. In Table 1, group A has lower MSE and KLD loss than group B, but its PCC is also lower.

To solve this problem, we train the model to maximize PCC directly. Hence, the loss function is given by:

$$L_{\text{PCC}} = -\frac{\sum_{n=1}^{N}(y^n - \bar{y})(\hat{y}^n - \bar{\hat{y}})}{\sqrt{\sum_{n=1}^{N}(y^n - \bar{y})^2}\sqrt{\sum_{n=1}^{N}(\hat{y}^n - \bar{\hat{y}})^2}} \quad (5)$$

where $\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y^n$ and $\bar{\hat{y}} = \frac{1}{N}\sum_{n=1}^{N}\hat{y}^n$. Since N is fixed for every batch, L_{PCC} is differentiable with respect to y^n, which means we can apply back propagation to train the network. To the best of our knowledge, we are the first team to adopt this training objective.

Group	A			B		
Gold Score	3	4	5	3	4	5
$P(S=0)$	0.05	0.05	0.05	0.15	0.05	0.1
$P(S=1)$	0.05	0.05	0.05	0.3	0.2	0.1
$P(S=2)$	0.15	0.1	0.05	0.25	0.3	0.2
$P(S=3)$	0.5	0.35	0.0	0.1	0.25	0.3
$P(S=4)$	0.15	0.4	0.1	0.1	0.1	0.2
$P(S=5)$	0.1	0.05	0.7	0.1	0.1	0.1
$E[S]$	2.95	3.15	4.2	2.0	2.45	2.7
	MSE	KLD	PCC	MSE	KLD	PCC
	0.455	**1.966**	0.931	2.90	6.91	**0.987**

Table 1: Example of lower MSE and KLD not indicating higher PCC.

4 Evaluation

4.1 Data

Dataset	Pairs
Training	22,401
Validation	5,601

Table 2: Training and validation Data sets (STS 2012-2016 and SICK).

We gather dataset from SICK (Marelli et al., 2014) and past STS across years 2012, 2013, 2014, 2015, and 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016) for both cross-lingual and monolingual subtasks. We shuffle and split them according to the ratio 80:20 into training set and

validation set, respectively. Table 2 indicates the size of training set and validation set. All non-English sentence appearing in training, validation, and test set are translated into English with Google Cloud Translation API.

4.2 Experiments

In the experiment, the size of output of GRU is set to be $H = 200$. We use ADAM algorithm to optimize the parameters with mini-batches of 125. The learning rate is set to 10^{-4} at the beginning and reduced by half for every 5 epochs. We trained the network for 15 epochs.

Word embeddings In Table 3, we demonstrate that the system performs better with pretrained word vectors (WI) than randomly initialized (RI).

	D	PCC on validation set
RI	50	0.7904
	300	0.8091
WI	50	0.7974
	300	0.**8174**

Table 3: System performance with different dimensions of word embeddings, using either randomly initialized or pre-trained word embedding.

Loss function We display performances with systems optimized with KLD, MSE, and PCC. It shows that when using L_{PCC} as the training objective, our system not only performs the best but also converges the fastest. As shown in Table 4 and Figure 2.

Loss function	PCC
L_{KLD}	0.6839
L_{MSE}	0.7863
L_{PCC}	**0.8174**

Table 4: Influence of different loss objectives on the system performance measured using PCC on our validation set.

4.3 Final System Results

We tune the model on validation set, and select the set of hyper-parameters that yields the best performance to obtain the scores of test data. We report the official provisional results in Table 5. There is an obvious performance drop in track4b, which happens to all teams. We hypothesize that the sentences in track4b (en_es) are collected from a special domain, due to the fact that the number of out-of-vocabulary words in track 4b is many times more than that in other tracks.

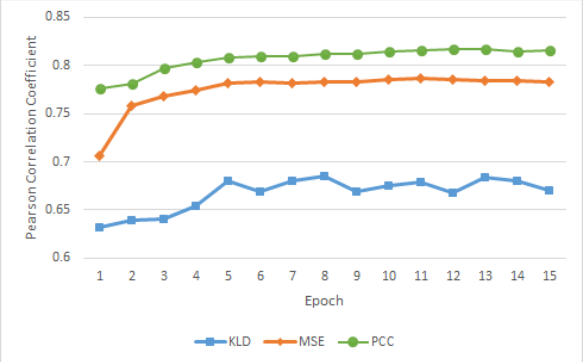

Figure 2: Performance of different loss functions

Track	PCC	mean	median	max
Primary	0.6171	0.66	0	28
1	0.6821	0.53	0	3
2	0.6459	0.50	0	3
3	0.7928	0.35	0	4
4a	0.7169	0.35	0	4
4b	0.0200	2.54	2	28
5	0.7927	0.36	0	4
6	0.6696	0.33	0	5

Table 5: Final system results and statistics of the number of OOV words within a pair

5 Conclusion

We propose a simple neural-based system with a novel means of optimization. We adopt a simple neural network with surface features which leads to a promising performance. We also revise several popular training objectives and empirically show that optimizing directly on Pearson's correlation coefficient achieved the best scores and perform competitively on STS-2017.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 252–263. http://www.aclweb.org/anthology/S15-2045.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 81–91. http://www.aclweb.org/anthology/S14-2010.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, San Diego, California, pages 497–511. http://www.aclweb.org/anthology/S16-1081.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, pages 385–393. http://www.aclweb.org/anthology/S12-1051.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 32–43. http://www.aclweb.org/anthology/S13-1004.

Ahmet Aker, Frederic Blain, Andres Duque, Marina Fomicheva, Jurica Seva, Kashif Shah, and Daniel Beck. 2016. Usfd at semeval-2016 task 1: Putting different state-of-the-arts into a box. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 609–613. http://www.aclweb.org/anthology/S16-1092.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Tomáš Brychcín and Lukáš Svoboda. 2016. Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 588–594. http://www.aclweb.org/anthology/S16-1089.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. http://www.aclweb.org/anthology/D14-1179.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*. Association for Computational Linguistics, pages 13–18.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*. Citeseer, pages 45–52.

Wael H Gomaa and Aly A Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications* 68(13).

Hua He, John Wieting, Kevin Gimpel, Jinfeng Rao, and Jimmy Lin. 2016. Umd-ttic-uw at semeval-2016 task 1: Attention-based multi-perspective convolutional neural networks for textual similarity measurement. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1103–1108. http://www.aclweb.org/anthology/S16-1170.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49(2):265–283.

Peng Li and Heng Huang. 2016. Uta dlnlp at semeval-2016 task 1: Semantic textual similarity: A unified framework for semantic processing and evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pages 584–587. https://doi.org/10.18653/v1/S16-1088.

Matthias Liebeck, Philipp Pollack, Pashutan Modaresi, and Stefan Conrad. 2016. Hhu at semeval-2016 task 1: Multiple approaches to measuring semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 595–601. http://www.aclweb.org/anthology/S16-1090.

Dekang Lin et al. 1998. An information-theoretic definition of similarity. In *ICML*. Citeseer, volume 98, pages 296–304.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval* 5(2–3):103–233.

Barbara Rychalska, Katarzyna Pakulska, Krystyna Chodorowska, Wojciech Walczak, and Piotr Andruszkiewicz. 2016. Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pages 602–608. https://doi.org/10.18653/v1/S16-1091.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, pages 441–448. http://aclweb.org/anthology/S12-1060.

Arafat Md Sultan, Steven Bethard, and Tamara Sumner. 2016. Dls$@$cu at semeval-2016 task 1: Supervised models of sentence similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pages 650–655. https://doi.org/10.18653/v1/S16-1099.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1556–1566. http://www.aclweb.org/anthology/P15-1150.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37:141–188.

Ngoc Phuoc An Vo, Simone Magnolini, and Octavian Popescu. 2015. Fbk-hlt: An application of semantic textual similarity for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*. volume 15, pages 231–235.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, pages 441–448. http://www.aclweb.org/anthology/S12-1060.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 985–992.