DEIM Forum 2018 A1-3

Detection of mergeable Wikipedia articles based on overlapping topics

Renzhi Wang [†]　Mizuho Iwaihara [‡]

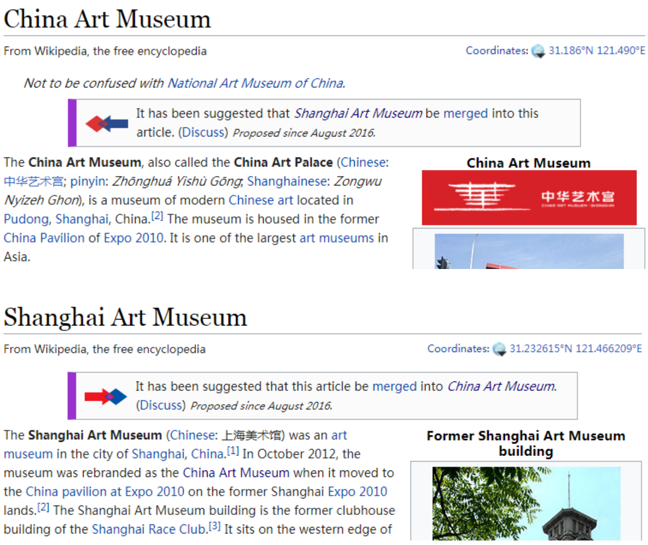
Graduate School of Information, Production and Systems, Waseda University
Kitakyushu 808-0135, Japan
E-mail: [†] ouninnyui.ips@asagi.waseda.jp, [‡] iwaihara@waseda.jp

Abstract Wikipedia is the largest online encyclopedia, in which articles are edited by different volunteers with different thoughts and styles. Sometimes two or more articles' titles are different but the themes of these articles are exactly the same or strongly similar. Administrators and editors are supposed to detect these article pairs and determine whether they should be merged together. In this paper, we propose a method to automatically determine whether an article pair should be merged together. According to Wikipedia Guidelines for article merge, in the duplicate case, the article pairs are covering exactly the same contents. In the overlap case, the articles pairs are covering related subjects that have a significant overlap. The content of an overlapped part is similar but the words in the pair are probably different, so methods that exploit semantic relatedness are necessary. To deal with this problem we consider both term co-occurrence similarity and semantic relatedness. We propose combination of multiple embedding results and rebuilding word vectors for evaluating semantic relatedness. We also deal with overlap cases by computing Jaccard distance between article pairs. Our experiments show that our method performs better than existing embedding methods.

Keyword Word embedding, Mergeable article, Wikipedia, Text mining

1. Introduction

1 Wikipedia articles are edited by various volunteers from all over the world. Each article in Wikipedia identifies a clear concept. Due to diverse culture and cognition backgrounds, one concept may be written in various styles by different editors in different articles. Administrators and editors need to merge these articles to avoid confusing readers and remove duplications. As stated in the Wikipedia guidelines for merge[11], there are four reasons to merge articles: duplicate, overlap, text, and context. The duplicate and overlap reasons are about identical content. If two or more articles are exactly the same content or they have a large overlap, then they should be merged. The text reason means one article is very short and has little content, so it should be merged to a larger inclusive topic. The context reason is that if a short article needs many background materials, it should be merged with a broader article. Currently the Wikipedia article merge task is done by human editors after discussion. For example, the articles "China Art Museum" and "Shanghai Art Museum" are suggested to be merged together, because an identical museum used to be called Shanghai Art Museum was rebranded as the China Art Museum in October 2012. The museum is actually the same museum, but there are two articles in Wikipedia, and the contents of these articles have a large overlap. It is like near duplicate text detection problem. But we also need to consider the semantic similarity between article pairs. In Wikipedia, near duplicate text detection is necessary for copyright enforcement and help version management.



China Art Museum

From Wikipedia, the free encyclopedia　Coordinates: 31.186°N 121.490°E

Not to be confused with National Art Museum of China.

It has been suggested that *Shanghai Art Museum* be merged into this article. (Discuss) *Proposed since August 2016.*

The **China Art Museum**, also called the **China Art Palace** (Chinese: 中华艺术宫; pinyin: *Zhōnghuá Yìshù Gōng*, Shanghainese: *Zongwu Nyizeh Ghon*), is a museum of modern Chinese art located in Pudong, Shanghai, China.[2] The museum is housed in the former China Pavilion of Expo 2010. It is one of the largest art museums in Asia.

China Art Museum

Shanghai Art Museum

From Wikipedia, the free encyclopedia　Coordinates: 31.232615°N 121.466209°E

It has been suggested that this article be merged into *China Art Museum*. (Discuss) *Proposed since August 2016.*

The **Shanghai Art Museum** (Chinese: 上海美术馆) was an art museum in the city of Shanghai, China.[1] In October 2012, the museum was rebranded as the China Art Museum when it moved to the China pavilion at Expo 2010 on the former Shanghai Expo 2010 lands.[2] The Shanghai Art Museum building is the former clubhouse building of the Shanghai Race Club.[3] It sits on the western edge of

Former Shanghai Art Museum building

Figure 1 Mergeable Wikipedia Articles

For a large collection of documents, comparing a query with every document in the collection is too costly, so conventional approaches mainly focus on how to select small candidates efficiently. After obtaining small candidates, two documents are compared mainly by term co-occurrence similarity. Previous researches[1][2][3][4][7] focus on dealing with large datasets. On the other hand, our goal is to detect the articles pairs which are exactly duplicated or have a significant overlap. Our candidate set is easily to be selected by Wiki search, so we put more emphasis on semantic similarity, because in Wikipedia, different editors often use different words in writing an identical article, although their intensions are basically the same. Term co-occurrence similarity is not fit for the case of diverse wordings with same intension.

In this paper, besides overlap, we mainly consider the semantic similarity, to extract the semantic meaning of articles. Because the nonoverlap part of the mergeable article pair can be regarded as the complement of the overlap, their semantic similarity is also significant. We adopt the popular word embedding method, word2vec[8][9]. The difficulty of our task is that, the known pre-trained embedding result is based on a very large corpus. Compared with such a corpus, our target dataset is just several articles from a part of Wikipedia, so the distribution of word occurrence can be distinctively skewed. So directly using pre-trained embedding causes undesirable results such as words in our target dataset which have a specific meaning. Directly using our targetdataset to train a new embedding result is also undesirable, because compared with large corpora, our dataset is too small to train a good embedding result. To solve this problem, we propose utilizing transfer matrixes like translation matrix in [9] to combine multiple pre-trained embedding results, and we introduce a new loss function to fit for the target dataset.

Our approach is motivated by transductive transfer learning[6]. The definition of transductive transfer learning is that the source domain (Ds) and source domain task(Ts) is given and the target domain(Dt) and target domain task(Tt) is the goal. Here Ts is equal to Tt but Ds is not equal to Dt. The transductive transfer learning methods want to use the knowledge in the source domain and source domain task to improve the prediction function in the target domain and target domain task. Usually, the source domain task has large labeled data, while the target domain task has only a limited label dataset. In our case, the pre-trained embedding results are the source domain and source task. Our mergeable articles dataset is the target domain. We propose a new loss function to improve the embedding results in our mergeable article dataset.

Our experiments on real Wikipedia mergeable articles show that our method predicts better than both local embeddings trained over just the target dataset and global embeddings trained over large corpora. As criteria for mergeable articles in Wikipedia, we utilize both Jaccard distance and semantic similarity by word2vec in measuring overlaps.

The rest of the paper is organized as follows: Section 2 introduces related work on related tasks. Section 3 shows our proposing method. In Section 4 we describe our datasets in detail, explain our experimental process and evaluation results. Section 5 is a conclusion.

2. Related Work

For near duplicate text detection task, a variety of signature selecting methods, encompassing scalability, have been proposed. Previous researches [2][3][4][7] separately proposed shingling-based, windowing-based, simhash-based algorithms to detect near duplicate texts. But these methods only exploit co-occurring terms, where semantic relatedness is not considered. These methods cannot handle texts that use a large number of different terms but expressing the same topic.

Recent researches consider incorporating semantic information into document signatures. Alonso et al. [1] considered TF-IDF weighting in their signature algorithm, to reflect certain semantic information.

Word embeddings are becoming an effective way to represent words by relatively low-dimensional vectors, where semantic relatedness is easily given by cosine similarity of two word vectors. To integrate different embedding results, [10] utilized convolutional neural networks. In their model the target dataset is just for classification, not participating in training embedding results.

3. Proposed Method

Recently, the word2vec model has become the most popular embedding model [8]. Word2vec assumes two language models, Continuous Bag of Words (CBOW) and Skip-gram. The CBOW language model assumes that context words' vectors should predict the target word. While the skip-gram model assumes that the target word should predict the context words. Based on these assumptions, they define objective functions as products of all target words' predicted probabilities. To perform training efficiently over large datasets, word2vec uses Huffman tree to maximize the objective function. Here the objective functions are defined as:

$Objective = \sum_{w \in C} logP(w|context(w))$------CBOW (1)

$Objective = \sum_{w\in C} logP(context(w)|w)$--Skip-gram (2)

After training, we can obtain distributed word vector representations, which will be used to compute similarities between article pairs.

Our goal needs to deal with a small training dataset of mergeable articles. To combine pre-trained embedding results, we utilize transfer matrixes to fit each embedding result. We also define the sum of all the embedding results multiplied by transfer matrixes as the final embedding result. The formula is as bellow.

$$E_f = \sum_{i=1}^{n} E_i \cdot T_i$$

Here, $\mathrm{E_i}$ is the pre-trained embedding result and $\mathrm{T_i}$ is the transfer matrix, *n* is the count of pre-trained embedding results and $\mathrm{E_f}$ is the final embedding results.

To fit for the target dataset, we define a new loss function. As the original word2vec model assumed, we also suppose that context words can predict a target word. In embedding space, this assumption can be regarded as the average of context words should be the closest to the target word and the average of context words should be far away from the other words.

Based on this assumption, we define the objective function as follow:

$Objective = \sum_{w\in C} Dis(context(w), w)$---------------------(3)

Here Dis function is the distance between the sum of context word vectors and target word vector. C is the corpus. Here the distance function can be any reseanable distance such as Manhattan distance, Euclidean distance, cosine similarity and so on. In our case, we use Euclidean distance as our Dis function. The different between our objective function and CBOW is that we use Euclidean distance as our Dis function. The advantage of using Euclidean distance is that for small datasets we do not need to build a softmax layer (in the word2vec model that is a Huffman tree) to compute the probability, instead we can directly compute the Euclidean distance between the context word vector and target word vector. The difference between our method and the original word2vec model is that we want to minimize this distance objective function, but not the product of the predicted probabilities of all the target words. To minimize the objective function, we can use stochastic gradient descent (SGD) in computing the transfer matrix. We can train to obtain the final result by SGD. In our experiments, we use tensorflow to achieve the optimization. We define the pre-trained embedding results as the placeholder and define transfer matrixes as variables. Then we apply gradient decent optimizer to minimize the loss function.

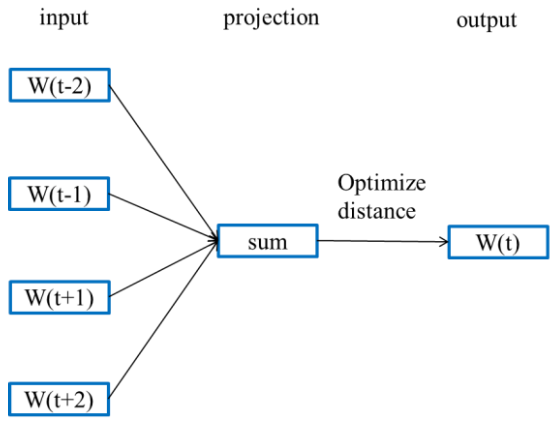


Figure 3: Proposed assumption

Another important difference between our method and previous work is that our objective function is trained on the target corpus. Pre-trained embedding results are based on well-known datasets, such as Wikipedia and Google News, which can have disagreements in vocabularies and distributions from the target dataset. Because global embedding results were trained based on the objective function over global dataset (Wikipedia, Google News), our final embedding results were trained over our target dataset so it is expected that the new embedding result can fit the target dataset better than pre-trained embedding results.

After we obtain the target-final word vectors, we define document vectors as the sum of all the word vectors in the document. We compute the cosine similarity between article pairs as their semantic similarity.

Besides the word2vec model, we also utilize Jaccard distance to measure the overlap between two articles. Jaccard distance is suitable for measuring duplicates and overlaps. We define Jaccard distance between two articles as below, here the word set in articles are after removing stopwords.

$$Jaccard\ distance(A, B) = \frac{(word\ set\ in\ A)\ intersect\ (word\ set\ in\ B)}{(word\ set\ in\ A)\ union\ (word\ set\ in\ B)}$$

We do not use the square root over the denominator because the square root over the denominator is not normalized and reflect the size of overlaps.

In our task to detect mergeable articles, we combine these two features to deal with all the criteria of article merge. We utilize linear combination to combine these two features to predict the most probable article pair that should be merged.

We show an example of how embedding model and Jaccard distance fit for the criteria of article merge, the articles "China Art Museum" and "Shanghai Art Museum". In the Wikipedia discussion page (https://en.wikipedia.org/wiki/Talk:China_Art_Museum#Proposed_merge_with_Shanghai_Art_Museum) the editors gives some reasons that "article on former museum could be merged into history section of current museum." "The building may be new but most of the collection will be the same." We can find the reason above from text.

The first paragraphs of both articles describe the same entities including the museum is in Shanghai and in October 2012 the museum was rebranded as China Art Museum, the museum is housed in the former China Pavilion of Expo 2010. The first paragraphs of both articles are short but describe three identical facts. The overlap of these two paragraphs is high. The first segments in paragraph "History" of two articles also describe the same key words like "Nanjing Road", "Shanghai Race Club" and area sizes. These two segments are quite similar with each other and both short. To handle this case, we think about Jaccard distance is suitable to measure the similarity between articles.

The other paragraphs in two articles are not the same facts, but content still related. They both describe the famous artworks and events, but due to different times, the artworks and events are not the same, so the words in these two parts will not be same, but they are still related with each other. As they both describe the famous artworks and events, we expect they have strong semantic relatedness. To handle this case, we use embedding model to measure the semantic relatedness. Thus we deal with the criteria of article merge by separately compute the Jaccard distance and semantic relatedness and combine the two similarities together.

We show the detail data in my experiments. If we just see the Jaccard distance, the article "People's Square" is most similar with the "Shanghai Art Museum", the Jaccard distance is 0.189, larger than the Jaccard distance between "Shanghai Art Museum" and "China Art Museum" 0.160, that is because the article "People's Square" is short, so Jaccard distance is high. If we just consider the semantic relatedness between articles, for "Shanghai Art Museum" the most related article is "Shanghai Museum", that is because they are built near each other, and they exhibit similar artworks. So embedding model gives strong related between these two articles. But when we combine Jaccard distance and semantic relatedness together, the "Shanghai Art Museum" and the "China Art Museum" become the most mergeable pair. We both consider about overlap and semantic relatedness. This example shows how our propose method deal with criteria of article merge.

4. Experiments

We extracted 5460 pairs of articles in total which are suggested to be merged together from the category page (https://en.wikipedia.org/wiki/Category:All_articles_to_be_merged). These articles in Wikipedia are labeled by "It has been suggested that this article be merged into …". For each of the mergeable articles, we searched the article title by Wikisearch, and downloaded top 20 results. We insert the correct answer (the other article of the mergeable pair) into the search results, if the correct answer is not already in the search result, so there can be 20 or 21 articles in the candidate set which includes the correct answer. The corpus totally includes 114574 articles.

Given one article, our algorithm will select one article which should be merged together from the candidate set (select 1 from 21). As baseline models by single features, we evaluate TF-IDF, Jaccard Distance and simhash on our dataset. For embedding results, we evaluate three pre-trained embedding results in Table 1 and we directly train embeddings on the target mergeable articles dataset. Table 2 shows the results.

Table 1: Details of pre-trained embeddings

dataset	Word count	Dataset size	Training method
Wikipedia	400K vocabulary	6 billion tokens	Glove
Google News	3M vocabulary	100 billion tokens	Skip-gram
Common Crawl	2.2M vocabulary	840 billion tokens	Glove

Table 2: Single model result

Single Method	accuracy
TF-IDF	0.024
Jaccard distance	0.436
Simhash	0.070
Embedding(Wikipedia)	0.527
Embedding(Google News)	0.537
Embedding(Common Crawl)	0.534
Directly train embedding result on dataset	0.435

For combining pre-trained embedding results, we compare different methods for combining the multiple embedding results. We adopt linear combination, Autoencoder combination[5] and our proposed method. These methods are all unsupervised, for linear combination, each dimension in final embedding is the average of dimensions in every pre-trained embedding result. The results are shown in Table 3

Table 3: Combination model result

Combining embedding result(Common Crawl and Google News)	accuracy
Linear combination	0.535
Autoencoder combination	0.536
Transfer matrix combination	0.539

The combined embedding results above are just comparing semantic similarities. When we add Jaccard distance that is expected to measure overlaps, duplicates and length of articles pairs, the results are expected to be improved. The results are shown in Table 4.

Table 4: Combining features

Features	accuracy
Embedding(Google News)+ Jaccard	0.608
Embedding(Common Crawl + Google News)+ Jaccard	0.613

Here, we combine features by linear combination. We just use the semantic similarity plus Jaccard distance as the final similarity.

To test our method on various overlaps of article pairs, we divide our dataset into three subsets, with low, middle and high overlaps. The low overlap is the pairs that have less than 20 co-occurring words. The middle overlap is the pairs that have between 21 and 60 co-occurring words, and the high overlap is the pairs having more than 60 co-occurring words. The overlap distributions of our datasets are shown in Figure 2. The results of the compared methods over the subsets are shown in Table 5.

Table 5: Accuracy results over different overlaps

Method|Overlap	[0,20]	(20,60]	(60,+∞]
Pair count	192	2204	3064
Jaccard	0.375	0.477	0.385
Embedding(Wikipedia)	0.188	0.344	0648
Embedding(Google News)	0.192	0.370	0.649
Embedding(Common Crawl)	0.203	0.363	0.644
Transfer matrix combination(Google News + Common Crawl)	0.188	0.375	0.660
Embedding(Google News)+ Jaccard	0.297	0.467	0.698
Embedding(Common Crawl + Google News)+ Jaccard	0.296	0.480	0.704

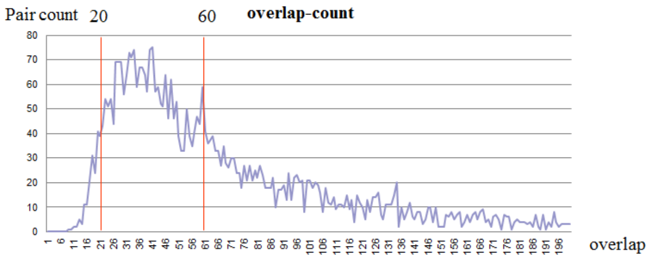


Figure 2 distribution of word overlapping

From the results shows in Table 5, we observe that the overlap affects the accuracy results of the Jaccard distance and embedding-based semantic relatedness measures. As we employ linear combination of Jaccard distance and semantic relatedness, we think about overlap size and article length as parameters to adjust the weight between Jaccard distance and embedding-based semantic relatedness. From Table 5, we find in the low overlap size Jaccard distance perform better, and in high overlap size the embedding based methods perform better.

Compared with short article pair, long article pair has more words so they should probably have high overlap. For the same reason, short article pair should probably have low overlap. So our principle is that for long article pair we will give more weight on embedding based methods and for short article pair we will give more weight on Jaccard distance. But for one short article and one long article pair, compared with both short article pair, the overlap size is expected not increase much, but the total article length increases much, so we prefer to put more weight on semantic relatedness. For one short article and one long article pair, compared with both long article pair, the overlap size decrease not so much, and the total article length decrease much, so compared with both long article pair, we prefer to put more weight on Jaccard distance.

We define two functions to measure the effective of article length over above the principle. The formula is as below:

$$similarity = F(overlap\_size) * jaccard\ distance + G(overlap\_size) * semantic\ relatedness$$

Here F() and G() are two functions to measure the weight with document length. Our samples are article pair, we normalize the F() and G() by divide the product of article length. To fit for our assumption above, we set F as a decrease function and set G an increase function. We try some popular functions to modify the weight, and the result shows as below. And we set our normalize function as $norm(x) = x \cdot \frac{1}{length(A)*length(B)}$, different from Jaccard distance normalize function. In F() the α is a static weight.

Table 6: measure weight function

F()	G()	accuracy
$norm(\frac{\alpha}{1+e^{x}})$	$norm(\frac{1}{1+e^{-x}})$	0.535
$norm(\frac{\alpha}{x})$	$norm(x)$	0.207
$norm(\frac{\alpha}{\log(x+1)})$	$norm$(Log(x+1))	0.206
$norm(\arctan\left(\frac{\alpha}{x}\right))$	$norm(\arctan(x))$	0.389

5. Discussion

From the results, we can find that the result of the word2vec-based methods is better than the TF-IDF, Jaccard-Distance and simhash-based methods. The reason is that while our goal is to find mergeable articles pair, simhash focuses on literal similarity. TF-IDF is affected by the datasets, performing worse in our dataset. Jaccard distance performs better than TF-IDF and simhash, because certain mergeable articles have a large overlap, making the Jaccard method produces high precisions.

From Table 3, we can see the embedding methods perform better than the conventional methods. It proves embedding methods fit for this task.

We compared four single embedding methods. The embedding model directly trained on the target dataset performs worst as we expected, since the target dataset is smaller than other pre-trained models and our combined model. The single embedding result trained over Wikipedia is not the best in the single embedding results. It can be explained as the corpus of Wikipedia is the smallest in the training datasets. The combined embedding methods are mostly better than the single embedding results. It is because the combined embedding supports more cases than single embedding methods. Usually, more cases yield a high precision. Another reason is that two different results will reduce the final vector bias of the words in target dataset, although the model is the same with the word2vec model, whose variance is stable. When we add a new different dataset, the bias will decrease. That could improve our result. The last reason is that our objective function is more adopted to a new particular target dataset, which is not reflected on embedding results trained over general large corpora.

For the combination methods, linear combination and autoencoder combination are totally unsupervised, while transfer matrixes method reflects given target datasets.

From Tables 3 and 4, we can find that when we combine an embedding result and Jaccard distance, the result can be significantly improved. It is because embeddings can evaluate semantic similarity well and Jaccard distance can evaluate overlaps and duplicates well. They fit well for the criteria of mergeable articles. Combination of these two features is expected to achieve a better result. We can also see combining multiple pre-trained embedding results is better than directly using only one pre-trained embedding result.

From Table 5 we can find the Jaccard distance perform best in the low overlap pairs, while embedding-based methods perform better in the high overlap pairs. That can be because the article pairs with low overlap are short, so Jaccard distance is much more important. Embedding-based methods performs better on pairs having high overlap because those article pairs are usually longer, so semantic similarities of nonoverlapping parts give more information.

We also try to measure the weight of Jaccard distance and semantic relatedness by the overlap size and article length, but the results are not improved. That's may because the alpha parameter is hard to determine. Also we find the function sigmoid and arctan gives better results. It may because sigmoid function and arctan function output the normalized results again.

6. Conclusion and future work

In this paper, we proposed a combination method of multiple embedding results. We consider not only term co-occurrence similarity but also semantic similarity between article pairs. We discussed the differences between pre-trained large datasets and target dataset, and introduced a new objective function. This objective function can train a model more fitted to a particular target dataset, reducing the bias of the global model. Focusing on detecting mergeable article pairs, we discussed combining pre-trained embedding results for evaluating sematic similarity and Jaccard distance for evaluating overlaps and duplicates. Combination of these two features shows around 10 percent improvement in accuracy, giving the best results. In the future work, consider combining different embedding result by transfer matrixes may not be the best choice, we can have a try on neural network based method such as Convolutional Neural Network (CNN) or Long-Short Term Memory Network (LSTM). These network structures can detect more information in the context. Compared with transfer matrixes, CNN can detect relationship between words such as in phrases. In transfer matrixes the context window must be fixed but LSTM can detect relationships between target words with long context. We will try to build new network structure based on these two neural network structures.

Reference

[1] Alonso, O., Fetterly, D., & Manasse, M. (2013, December). Duplicate news story detection revisited. In *Asia Information Retrieval Symposium* (pp. 203-214). Springer, Berlin, Heidelberg

[2] Broder, A. Z. (1997, June). On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings* (pp. 21-29). IEEE.

[3] Broder, A. Z., Glassman, S. C., Manasse, M. S., & Zweig, G. (1997). Syntactic clustering of the web. *Computer Networks and ISDN Systems*, *29*(8-13), 1157-1166.

[4] Charikar, M. S. (2002, May). Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing* (pp. 380-388)

[5] Ng, A., 2011. Sparse autoencoder. *CS294A Lecture notes*, *72*(2011), pp.1-19.

[6] Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), pp.1345-1359.

[7] Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003, June). Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 76-85)

[8] Tomas M., Ilya S., Kai C., Greg C., Jeffrey D.: Distributed Representations of Words and Phrases and their Compositionality, NIPS '13, Pages 3111–3119 (2013)

[9] Tomas M., Kai C., Greg C., Jeffrey D.: Efficient Estimation of Word Representations in Vector Space, ICLR '13 Proceedings of Workshop at International Conference on Learning Representations (2013)

[10] Zhang, Y., Roller, S., & Wallace, B. (2016). Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. *arXiv preprint arXiv:1603.00968*

[11] https://en.wikipedia.org/wiki/Wikipedia:Merging#Reasons for merger