A Skill Term Normalization Method for Recruitment Webpages Combining Literal and Contextual Similarity (Postprint)

Authors: Sun Yu, Jiang Jinde

Date: 2023-10-08T00:00:00+00:00

Abstract

[Purpose/Significance] To address the problem of numerous spelling errors in English skill terms in recruitment webpage texts, a method for normalizing skill terms in recruitment webpages is proposed.

[Method/Process] Combining literal similarity and contextual similarity, the similarity of skill terms is measured to form a network of similar skill terms, thereby normalizing skill terms in recruitment webpage texts.

[Results/Conclusion] One week of job posting information for computer-related positions was obtained from 51job, a mainstream domestic recruitment website, and the proposed method was used to normalize English skill terms in recruitment webpages. Experimental results show that the proposed method can automatically, accurately, and quickly normalize skill terms in recruitment webpage texts.

Full Text

A Skill Vocabulary Normalization Method for Recruitment Webpages Combining Literal and Context Similarity

Sun Yu[1], **Jiang Jinde**[2]

Abstract

This paper proposes a skill vocabulary normalization method for recruitment webpages to address the problem of numerous English skill word spelling errors in recruitment webpage texts. The method combines literal similarity and context similarity to measure the similarity of skill words, forming a similar skill word network to normalize skill words in recruitment webpage texts. One week'

s worth of computer job postings was obtained from the domestic mainstream recruitment website 51job to evaluate the proposed method. Experimental results demonstrate that the method can automatically, accurately, and quickly normalize skill vocabulary in recruitment webpage texts.

Keywords: recruitment webpage; skill; lexical normalization
Classification Number: G202

In recent years, with the rapid development of higher education and the continuous expansion of enrollment in China, the difficulty for college graduates to find jobs and for enterprises to recruit talent has become a social concern. To some extent, this dual dilemma stems from the mismatch between talent cultivation in Chinese universities and social demand. Particularly in the information age, enterprise demand for talent changes rapidly, while university talent cultivation cycles are long and curriculum development lags behind, resulting in student training that deviates from actual needs. Therefore, in the rapidly developing information age, quickly and accurately understanding enterprise skill requirements for recruited positions is particularly important. With the popularization of the Internet, online recruitment has become the mainstream method for enterprise hiring. Recruitment webpages often contain specific descriptions of skill requirements for positions, reflecting current market demand for talent skills. Thus, analyzing recruitment webpage information to understand society' s skill demands for talent in specific fields is an effective approach.

Since recruitment webpages are unstructured texts, a series of natural language processing operations are required to obtain relevant structured skill information. However, unlike traditional carefully edited and revised texts, online recruitment texts are often written non-standardly. In particular, skill terms in certain domains are typically English words with many spelling errors, such as misspelling "Oracle" as "Orace" or "Linux" as "Liunx." The non-standard writing of skill words in recruitment webpage texts interferes with traditional natural language processing methods based on standardized texts. Therefore, before conducting skill requirement analysis on recruitment webpage texts, it is crucial to convert non-standard English skill words into standardized forms.

In recent years, some studies have attempted to analyze enterprise skill demands using online recruitment information [1-6]. However, these studies typically adopt manual methods for skill vocabulary normalization, which cannot adapt to the characteristics of rapid updates and large data volumes in recruitment webpages. Currently, research on automatic normalization of skill words in recruitment webpages remains limited. Literature [7] proposes a word vector clustering method for skill vocabulary normalization in recruitment texts, but this method does not consider that misspelled skill words usually have similar literal forms, and word vector models cannot generate accurate word vectors for low-frequency words, thereby affecting the effectiveness of skill vocabulary normalization.

Through careful observation of skill words in recruitment webpage texts, we find that misspelled words typically have similar literal forms and similar contextual skill terms. Therefore, we propose a method that combines literal and context similarity to measure skill word similarity, forming a similar skill word network to normalize skill words in recruitment webpage texts. Experiments demonstrate that our proposed method can automatically, accurately, and quickly normalize skill words in recruitment webpage texts.

1 Related Research

Lexical normalization is the process of grouping multiple words into an equivalence class and is an important step in many natural language processing tasks such as machine translation, named entity extraction, and information retrieval. These tasks process "clean" corpora that have been normalized, thereby reducing model complexity. As a key step in corpus preprocessing, lexical normalization has always attracted researchers' attention, especially with the explosive growth of social media text in recent years, making social media text normalization a research hotspot.

Early text normalization work mostly used the noise channel model. Literature [8] first applied the noise channel model to text normalization tasks, proposing a substring-edit-based noise channel model that models the probability of substring transformations to improve text normalization effectiveness. Literature [9] enhanced the error model in the noise channel model by incorporating phonetic similarity between words and learning rules to predict each character' s pronunciation, with predictions depending on adjacent characters in the word. However, this model is supervised and requires large amounts of annotated corpora for training.

Based on the noise channel model, current lexical normalization methods can be divided into three categories: spelling correction, sequence labeling, and machine translation. Spelling correction methods assume that the process of words becoming non-standard is independent, simplifying text normalization to word spelling correction. Sequence labeling approaches treat lexical normalization as a sequence labeling problem: first generating candidate normalized words for each word in the text, then using the Viterbi algorithm based on a language model to find the word sequence with maximum joint probability. Commonly used sequence models include Hidden Markov Models [10] and Conditional Random Fields [11]. Machine translation methods leverage the concept of word alignment to model one-to-many, many-to-one, and many-to-many mappings in non-standard-to-standard word relationships [12-13]. Both sequence labeling and machine translation methods are supervised approaches requiring large amounts of annotated training data, which demands substantial manual annotation effort. Therefore, unsupervised spelling correction methods have become a research focus.

Spelling correction methods mainly include two types: those based on word

form similarity and those based on context similarity. The most representative method based on literal similarity calculates edit distance between words [18] to represent word similarity. However, in social text, non-standard forms may differ significantly from standard forms. Literature [14] proposes a word similarity model for social text that uses phonetic spelling, word clipping, and other transformations to measure word similarity. Context similarity refers to the probability of different words appearing in similar contexts. Currently, word vectors trained by neural networks are commonly used [7, 15-17]. Specifically, literature [7] proposes using word2vec to represent skill words and their contexts for recruitment text skill vocabulary normalization.

Most current spelling correction methods target social network text. Compared with the diverse forms of social network words, skill words in recruitment webpage texts typically exhibit literal similarity. However, word vector methods for context are not suitable because word vectors more accurately represent high-frequency words, while low-frequency word vectors are inaccurate, affecting context-based similarity calculation. Our proposed method is an unsupervised spelling correction method specifically for recruitment webpage texts that does not require annotated data, adapting to the characteristics of rapid updates and large data volumes in recruitment webpages.

2 Skill Vocabulary Normalization Method Combining Literal and Context Similarity

Misspelled words in recruitment webpages typically have similar literal forms, such as misspelling "Oracle"as "Orace."However, using only literal similarity may misidentify non-misspelled words as misspelled. For example, "Radware" and "Hardware"have similar literal forms but are different words: "Radware"is a leading intelligent solution provider dedicated to ensuring fast, reliable, and secure delivery of network or web-based applications, while "Hardware" refers to computer system components. Through observation, we find that misspelled words usually have similar literal forms and similar contextual skill terms, whereas non-misspelled words with similar literal forms typically have different contextual skill terms. For instance, "Oracle" and "Orace" usually co-occur with "database" and "SQL," while "Radware" typically appears with "WebLogic," "Bea," and "Server," and "Hardware" appears with "monitor," "motherboard," "CPU," and "memory."

Therefore, to address the problem of numerous English skill spelling errors in Chinese recruitment webpage texts, we propose a skill vocabulary normalization method combining literal and context similarity. The overall process is shown in [Figure 1: see original paper]. The method consists of three main steps: preprocessing, calculating skill word pair similarity, and generating a similar skill word network. First, we perform preprocessing on the obtained recruitment webpage texts, then calculate the literal and context similarity of skill word pairs to form a similarity measure, and finally generate a similar skill word network based on the similarity measure for skill vocabulary normalization.

2.1 Preprocessing

Recruitment webpage texts are unstructured web documents containing not only required information such as skills but also substantial noise like advertisements, images, animations, irrelevant hyperlinks, scripting languages, and various tags. Therefore, we need to parse the web text using web text analysis tools to extract text structures related to recruitment analysis information. We then perform deduplication, part-of-speech tagging, and English case conversion on the obtained relevant text content. Since this paper aims to normalize English skill words in recruitment webpage texts, we filter out Chinese words and retain only English skill words. [Figure 2: see original paper] shows an example of preprocessing a recruitment webpage text, retaining all English words from the job requirements section, with each job posting webpage forming a corresponding job skill word text.

2.2 Calculating Skill Word Pair Similarity

To normalize skill words, we need to calculate similarity between skill words to determine the likelihood that two words represent the same skill. We measure skill word similarity from both literal and contextual aspects: when two words are more similar literally and contextually, they are more likely to be the same skill word.

2.2.1 Literal Similarity The most representative method based on literal similarity uses edit distance (ED) [18] to represent word similarity. Edit distance is the minimum number of insert, delete, and substitute operations required to transform one candidate skill term into another. Edit distance considers not only the number of identical characters between candidate skill terms but also their positional relationships. Typically, smaller edit distance indicates greater similarity between candidate skill terms. For example, the edit distance between "oracle" and "orace" is 1. Since edit distance does not consider the length of candidate skill terms themselves, we incorporate the lengths of both candidate skill terms into the edit distance to form normalized edit distance (NED), defined as follows:

$$\mathrm{NED}(w_i, w_j) = \frac{\mathrm{ED}(w_i, w_j)}{\mathrm{len}(w_i) + \mathrm{len}(w_j)} \tag{1}$$

In formula (1), $\mathrm{ED}(w_i, w_j)$ represents the edit distance between skill words w_i and w_j. According to the NED definition, smaller NED values indicate greater similarity between two skill words. When w_i and w_j are identical, their NED is 0. The NED value between "oracle" and "orace" is $\frac{1}{6+5} = \frac{1}{11}$.

Based on normalized edit distance, the literal similarity strSim between word w_i and word w_j is defined as:

$$\text{strSim}(w_i, w_j) = 1 - \text{NED}(w_i, w_j) \tag{2}$$

Formula (2) shows that when the normalized distance between two words is smaller, their literal similarity is larger, indicating the two words are more likely to be the same word. To avoid division by zero, we do not consider cases where w_i and w_j are completely identical.

2.2.2 Context Similarity Similar skill words usually have similar contextual skill words. Therefore, we can use the context of skill words to determine their similarity. Specifically, given word w_i, let D_i be the set of all job skill word texts containing skill word w_i, i.e., $D_i = \{D_{i1}, ..., D_{im}, ..., D_{in}\}$. We define two context similarity measures: conSetSim and conFreSim. The conSetSim is defined as:

$$\text{conSetSim}(w_i, w_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{3}$$

In formula (3), S_i is the contextual skill word set of w_i, where $S_i = \{w | w \in D_i \wedge w \neq w_i\}$. Similarly, S_j represents the contextual skill word set of w_j. Formula (3) shows that when skill word w_i and skill word w_j share more identical skill words in their contextual skill word sets, they are more similar and more likely to represent the same concept.

The conFreSim is defined as:

$$\text{conFreSim}(w_i, w_j) = \sum_{w \in S_i \cap S_j} \min \left(\frac{c(w, D_i)}{\sum_{w' \in S_i} c(w', D_i)}, \frac{c(w, D_j)}{\sum_{w' \in S_j} c(w', D_j)} \right) \tag{4}$$

In formula (4), $c(w, D_i)$ represents the number of times word w appears in the job context text set D_i of word w_i. Similarly, $c(w, D_j)$ represents the number of times word w appears in the job context text set D_j of word w_j. Formula (4) shows that conFreSim shares similarities with conSetSim in measuring word similarity through contextual word sets, but conFreSim also considers the frequency of each skill word in the contextual skill word sets.

Comprehensively considering both literal and context similarity measures, we form two final similarity metrics: strSim-conSetSim and strSim-conFreSim, defined as follows:

$$\text{strSim-conSetSim}(w_i, w_j) = \text{strSim}(w_i, w_j) \times \text{conSetSim}(w_i, w_j) \tag{5}$$

$$\text{strSim-conFreSim}(w_i, w_j) = \text{strSim}(w_i, w_j) \times \text{conFreSim}(w_i, w_j) \tag{6}$$

Formulas (5) and (6) show that when w_i and w_j have greater literal similarity and greater context similarity, the metrics strSim-conSetSim and strSim-conFreSim are larger, indicating that w_i and w_j are more likely to be the same skill word.

2.3 Generating Similar Skill Word Network

Based on the similarity of skill word pairs, we can generate a similar skill word network to find all skill words representing the same skill concept. Each vertex in the similar skill word network represents a skill word, and undirected edges between vertices represent skill word pairs with similarity greater than a predefined threshold. According to the generated undirected network, we find all connected components in the network, which represent sets of skill words denoting the same concept. The most frequently occurring skill word in each set is used to represent that set for skill vocabulary normalization. [Figure 3: see original paper] shows an example of a similar skill word network. In the network in [Figure 3: see original paper], there are three connected components forming three skill word sets: {websphere, webspere, websphare}, {visio, viso, vioso}, and {zibbix, zabbix, zabix}.

3 Experiments

To verify the feasibility and effectiveness of our proposed method, we crawled recruitment webpage data from the domestic mainstream recruitment website 51job (www.51job.com) to normalize skill words. 51job is a leading online recruitment service provider and one of China' s most influential talent recruitment websites. Following functional classifications, we selected the "Computer/Internet/Communication/Electronics" category on 51job to crawl data (data collection period: March 19-26, 2018) as our recruitment webpage corpus. After removing duplicate content, fully English pages, and pages without job requirements, we obtained 14,678 relevant recruitment webpages.

3.2 Experimental Steps and Evaluation Method

The experiment first preprocesses recruitment webpage texts, including using BeautifulSoup to locate and parse webpage content to obtain job skill requirement texts, using Jieba for part-of-speech tagging, converting English uppercase to lowercase, etc., ultimately retaining English skill words. We obtained 7,156 distinct English words. We then calculate the literal and context similarity for each pair of distinct skill words to form final similarity measures. We manually set a threshold of 7 to form the similar skill word network, find all connected components in the network to form skill word sets, and use the word with the highest frequency in each set as the normalized word for normalization.

The experiment manually annotates skill word pairs to determine whether they represent the same skill concept and uses the P@N method to evaluate correctly assessed skill word pairs, calculated as follows:

$$P@N = \frac{\text{Number of similar skill word pairs that are the same skill concept}}{\text{Total number of similar skill word pairs}} \tag{7}$$

3.3 Results

3.3.1 Evaluation of Skill Word Similarity Methods The experiment first evaluates the skill word similarity methods proposed in Section 3.2 through four comparison groups, as shown in .

Group 1 uses literal similarity and two types of context similarity to calculate word pair similarity separately. The results are shown in [Figure 4: see original paper]. As seen in [Figure 4: see original paper], among the three similarity calculations, the strSim method achieves the highest accuracy in the first 600 skill word pairs. However, after 600 pairs, the accuracy of strSim drops rapidly, falling below the two context-based methods conSetSim and conFreSim. Between the two context similarity methods, conSetSim performs better than conFreSim. The strSim method calculates word pair similarity based on literal form, which can accurately identify some misspelled word pairs but also produces errors. For example, skill words "spring" and "swing" have similar literal forms but represent different computer skill concepts. The conSetSim and conFreSim methods use word context to determine similarity, achieving relatively stable accuracy. Since conFreSim considers the frequency of contextual skill words, it can more precisely characterize skill word pair similarity and thus outperforms conSetSim.

Group 2 evaluates the hybrid method strSim-conSetSim that combines context similarity conSetSim and literal similarity strSim to assess whether it improves accuracy (see [Figure 5: see original paper]). As shown in [Figure 5: see original paper], the strSim-conSetSim method that combines literal and context similarity significantly improves accuracy and overall outperforms both the individual literal similarity strSim and individual context similarity conSetSim methods. This indicates that literal similarity calculates word pair similarity from the surface form, while context similarity calculates similarity from word context; the two complement each other to achieve better results.

Group 3 compares the method combining context similarity conFreSim and literal similarity strSim, namely strSim-conFreSim, to evaluate its impact on performance (see [Figure 6: see original paper]). As shown in [Figure 6: see original paper], the results are similar to Group 2: the strSim-conFreSim method combining literal and context similarity significantly improves accuracy and overall outperforms both individual literal similarity and individual context similarity methods. This further demonstrates that combining literal and context similarity yields better results.

Group 4 compares the two combined methods of literal and context similarity, namely strSim-conSetSim and strSim-conFreSim, using strSim as the base-

line method. The results are shown in [Figure 7: see original paper]. As seen in [Figure 7: see original paper], between the two combined methods, strSim-conFreSim performs slightly better than strSim-conSetSim. This result is consistent with the conclusion from Group 1 that conFreSim outperforms conSetSim. Compared with conSetSim, conFreSim considers the frequency of contextual skill words, enabling more precise word characterization and thus achieving higher accuracy.

3.3.2 Comparison with Other Methods Next, the experiment compares the best method from Section 3.3.1, strSim-conFreSim, with the method in literature [7], using strSim as the baseline method. Literature [7] uses neural network word vector methods to calculate word vectors based on skill word context for normalizing recruitment text skill words, which we abbreviate as the word2vecSim method. The experimental results are shown in [Figure 8: see original paper]. As seen in [Figure 8: see original paper], our proposed method outperforms the word2vecSim method, and even the strSim method outperforms word2vecSim. The main reasons are threefold: (1) word2vec uses neural networks to calculate word vectors. Although it considers context, word2vec only achieves good accuracy when skill words appear frequently; performance is poor when skills appear infrequently. (2) The context considered by word2vec includes not only skill words but also other non-skill words, which affects the utilization of skill words and thus cannot accurately generate skill word vectors. (3) word2vec does not consider the literal similarity of skill word pairs.

3.3.3 Actual Case Analysis lists three actual cases. In , the first word pair has high literal similarity and high context similarity, resulting in a high final strSim-conFreSim similarity value, indicating they are the same skill concept. The second word pair in has high literal similarity but actually represents two different skill concepts; their context similarity is not high, resulting in a low final strSim-conFreSim similarity, correctly identifying them as not the same skill concept. The third word pair in consists of two related skill words that therefore have high context similarity but very low literal similarity, correctly identifying them as not the same skill concept. This demonstrates that combining literal and context similarity yields more accurate skill word similarity than using either method alone.

3.3.4 Normalization Examples Finally, the experiment uses the similar skill word network to find connected components, forming several skill word sets, and adopts the word with the highest frequency in each set as the standardized word. lists some normalization examples. The results in show that our proposed method combining literal and context similarity can effectively normalize misspelled skill words in recruitment webpages.

4 Conclusion

Online recruitment information often contains specific descriptions of skill requirements for positions, reflecting current market demand for talent skills. Therefore, analyzing online recruitment information to understand society's skill demands for talent in specific fields is an effective approach. However, unlike traditional carefully edited and revised texts, online recruitment texts are often written non-standardly. In particular, skill descriptions in certain domains are typically in English with many spelling errors. The non-standard writing of skill words in online recruitment texts interferes with traditional natural language processing methods based on standardized texts. Therefore, before conducting skill requirement analysis on online recruitment texts, converting misspelled English skill words into standardized forms is particularly important.

We propose a method that combines literal and context similarity to measure skill word similarity, forming a similar skill word network to normalize skill words in recruitment webpage texts. Using one week' s worth of computer job postings from the domestic mainstream recruitment website 51job, we normalized English skill words in recruitment webpages. Experimental results demonstrate that our proposed method can automatically, accurately, and quickly normalize skill words in online recruitment texts. This enables job position skill requirement analysis and knowledge discovery, resolves the information asymmetry between employment knowledge supply and demand, helps universities and college students effectively utilize online employment information resources, assists university program managers in quickly understanding enterprise skill demands for professional talent, and provides intelligence decision support for developing talent cultivation programs that meet enterprise needs.

References

[1] WOWCKO I. Skills and vacancy analysis with data mining techniques[J]. Informatics, 2015, 2(4):31-49.
[2] KIM J Y, LEE C K. An empirical analysis of requirements for data scientists using online job postings[J]. International journal of software engineering and its application, 2016, 10(4): 161-172.
[3] Xia Huosong, Pan Xiaoting. Research on the Relationship Between Big Data Academic Research and Talent Demand Based on Python Mining[J]. Journal of Information Resources Management, 2017, 7(1): 4-12.
[4] Zhan Chuan. Analysis of Professional Talent Skill Requirements Based on Text Mining: A Case Study of E-commerce Major[J]. Library Tribune, 2017, 5(1): 116-124.
[5] Xia Lixin, Chu Lin, Wang Zhongyi, et al. Construction of Employment Knowledge Demand Relationships Based on Web Text Mining[J]. Document, Information & Knowledge, 2016, 169(1):94-101.
[6] Liu Ruilun, Ye Wenhao, Gao Ruiqing, et al. Research on Text Clustering Based on Big Data Job Requirements[J]. Data Analysis and Knowledge Discovery, 2017, 12(12): 65-72.

[7] LUO Q, ZHAO M, JAVED F, et al. Macau: large-scale skill sense disambiguation in the online recruitment domain[C]// IEEE international conference on big data. Piscataway: IEEE, 2015:1324-1329.
[8] BRILL E, MOORE R C. An improved error model for noisy channel spelling correction[C]// Meeting of the Association for Computational Linguistics. Piscataway: IEEE, 2000:286-293.
[9] TOUTANOVA K, MOORE R C. Pronunciation modeling for improved spelling correction[C]// Proceedings of annual meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2002:144-151.
[10] CHOUDHURY M, SARAF R, JAIN V, et al. Investigation and modeling of the structure of texting language[J]. International journal of document analysis & recognition, 2007, 10(3):157-174.
[11] LIU F, WENG F, WANG B, et al. Insertion, deletion, or substitution? normalizing text messages without pre-categorization nor supervision[J]. 2012, 15(2):71-76.
[12] AW A T, ZHANG M, XIAO J, et al. A phrase-based statistical model for SMS text normalization.[C]// International conference on computational linguistics and meeting of the Association for Computational Linguistics. New York: ACM, 2006: 17-21.
[13] PENNELL D L, LIU Y. A character-level machine translation approach for normalization of SMS abbreviations[C]// International conference on artificial intelligence. Piscataway: IEEE, 2011,20(2):974-982.
[14] COOK P, STEVENSON S. An unsupervised model for text message normalization[M]. Stroudsburg: Association for computational linguistics, 2009.
[15] SRIDHAR V K R. Unsupervised text normalization using distributed representations of words and phrases[C]// The workshop on vector space modeling for natural language processing. Piscataway: IEEE, 2015:8-16.
[16] Shi Zhenhui, Sha Ying, Liang Qi, et al. Research on Variant Word Normalization Based on Character-Word Combination[J]. Computer Systems & Applications, 2017, 26(10):29-35.
[17] Luo Yangen, Li Xiao, Jiang Tonghai, et al. A Uyghur Word Normalization Method Based on Word Vectors[J]. Computer Engineering, 2018(2):220-225.
[18] DAMERAU F J. A technique for computer detection and correction of spelling errors[J]. Communications of the ACM, 1964, 7(3):171-176.

Author Contributions:
Sun Yu: Proposed research ideas, conducted experiments, wrote the paper;
Jiang Jinde: Analyzed experimental data, revised the paper, provided theoretical guidance.

Note: Figure translations are in progress. See original paper for figures.