CVBed: Structuring CVs using Word Embeddings

Shweta Garg
Delhi Technological University
New Delhi, India
shweta_bt2k14@dtu.ac.in

Sudhanshu S Singh and **Abhijit Mishra** and **Kuntal Dey**
IBM Research, New Delhi and Bangalore, India
{sudsing3, abhijimi, kuntadey}@in.ibm.com

Abstract

Automatic analysis of curriculum vitae (CVs) of applicants is of tremendous importance in recruitment scenarios. The semi-structuredness of CVs, however, makes CV processing a challenging task. We propose a solution towards transforming CVs to follow a unified structure, thereby, paving ways for smoother CV analysis. The problem of restructuring is posed as a *section relabeling problem*, where each section of a given CV gets reassigned to a predefined label. Our relabeling method relies on *semantic relatedness* computed between section header, content and labels, based on *phrase-embeddings* learned from a large pool of CVs. We follow different heuristics to measure semantic relatedness. Our best heuristic achieves an F-score of 93.17% on a test dataset with gold-standard labels obtained using manual annotation.

1 Introduction

Automatic processing of curriculum vitae (CVs) is important in multiple real-life scenarios. This includes analyzing, organizing and deriving actionable business intelligence from CVs. For corporates, such processing is interesting in scenarios such as hiring applicants as employees, promoting and transitioning employees to new roles *etc.* For individuals, it is possible to add value by designing CV improvement and organization tools, enabling them to create more effective CVs specific to their career objectives as well as maintain the CVs easily over time. Hence, it is important to transform CVs to follow a unified structure, thereby, paving ways for smoother and more effective manual/automated CV analysis.

The semi-structuredness of CVs, with the diversity that different CVs exhibit, however, makes CV processing a challenging task. For example, a first CV could have sections *personal details, education, technical skills, project experience, managerial skills, others* and a second CV, equivalent to the first one, could have sections *about me, career objective, work experience, academic background, proficiency, professional interests*, in that order. Note that, some sections are equivalent (e.g., *personal details* and *about me*) in the two CVs, some sections are simply absent in some CVs (e.g., any equivalent of *others* that is present in the first CV, is missing in the second CV) and some sections in one CV is a composition of multiple sections in another CV (e.g., *proficiency* in the second CV is a combination of *technical skills* and *managerial skills* of the first). In real-life, the variations are high, and the solutions available today are far from perfect. Clearly, the problem at hand requires attention.

Multiple industrial solutions, such as Text Kernel[1], Burning Glass[2] and Sovren[3], have attempted to solve the problem at hand, and are offered as commercial products. Several researchers have also investigated the problem. Yu et al. (2005) proposed a hybrid (multipass) information extraction model to assign labels to block of CVs. Subsequent works, such as Chuang et al. (2009) and Maheshwari et al. (2010), also used multipass approaches, and feature-based machine learning techniques. Kopparapu (2010) suggested a knowledge-based approach, using section-specific keywords and n-grams. Tosik et al. (2015) found word embeddings to be more effective compared to word types and other features for CRF mod-

[1] https://www.textkernel.com
[2] http://burning-glass.com
[3] https://www.sovren.com

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 349–354,
Taipei, Taiwan, November 27 – December 1, 2017

els. Singh et al. (2010) and Marjit et al. (2012), amongst others, also proposed different solutions.

We use a phrase-embedding based approach to identify and label sections, as well as investigate the usefulness of traditional language resources such as WordNet (Miller, 1995) and ConceptNet (Liu and Singh, 2004). Empirically, our approach significantly outperforms other approaches.

2 Central Idea

As discussed earlier, CVs generally do not follow any predefined structure, and hence it would be hard to propose a deterministic (rule-based) solution for parsing and categorizing the sections of CV. This necessitates the application of statistical classification to map each section of the CV to section-labels chosen from an exhaustive list of predefined labels. Now, applying supervised classification for this task would require a large amount of manually labeled training data which is extremely time consuming. Our approach, on the other hand, is based on unsupervised learning where each label is chosen based on the *semantic relatedness* between the label and the section content (in terms of section-header and section-body). For example, a section titled "Academic qualifications" could be semantically closer to a predefined label "Education" than "Skills"; the section would thus be categorized under "Education". We propose two schemes for obtaining the semantic relatedness between section headers, bodies and the predefined labels (discussed in Section 3.2).

2.1 Scheme 1: Exhaustive Comparison

In the first scheme, we perform an exhaustive similarity comparison of all the words that appear in the given section of the test CV, with the label set. In this scheme, for each section extracted from the CVs, content words from the section headers and bodies are extracted and combined. A *lexical similarity* measure is computed, between each label in the label-set and each word extracted from the section. The average lexical similarity score for each label, with all the words in the section, is then computed. The label with the highest average similarity score is selected as the winner label. The intuition behind this scheme is that, labels that share maximum lexical similarity with section have the maximum semantic relatedness with the section, hence, most appropriate.

Formally, let $L = \{l_1, l_2, ..., l_n\}$ be the set of available labels. Let $W = \{w_1, w_2, ..., w_m\}$ be the set of words present in a given section. Let $\sigma(w_i, l_j)$ represent the semantic similarity of word w_i with the label l_j. The average similarity $\lambda(l_j, W)$ of label l_j with the set of words W is computed as:

$$\lambda(l_j, W) = \frac{\sum_{i=1}^{m} \sigma(w_i, l_j)}{|W|} \quad (1)$$

The label selected as the winner, $\Lambda(L, W)$, is:

$$\Lambda(L, W) = \forall(j) max(\lambda(l_j, W)) \quad (2)$$

For computing semantic similarity σ, we use WordNet (Miller, 1995) *path* similarity (Leacock and Chodorow, 1998) and *Wu-Palmer* similarity (Wu and Palmer, 1994) and ConceptNet (Liu and Singh, 2004; Havasi et al., 2007) based similarity (Spagnola and Lagoze, 2011). The systems variants for these three similarity measures are, henceforth, referred to as PATH, WUP, CONCEPT. As expected, WordNet and ConceptNet offer limited coverage, resulting in many of the similarity scores as 0. We therefore propose another variant (referred to as EMBEDDING) where lexical similarity is the cosine similarity between the embeddings of the two input words. The word embeddings are learned using the training data consisting of 1179 CVs (detailed in Section 3.1) using the skip-gram approach (Mikolov et al., 2013a,b), implemented with the help of `gensim` package in python (Řehůřek and Sojka, 2010). The embedding dimension, min count, and window size were empirically set to 100, 5 and 4 respectively, and the vocabulary size turned out to be 7970.

2.2 Scheme 2: MultiEmbedding

In the second scheme, we employ MULTIEMBEDDING, a *representative word-cluster similarity based* approach. Here, instead of directly comparing the words appearing in the test data, we do the following. First, content words from section header and body are extracted and combined to form the set of words $W = \{w_1, w_2, ..., w_m\}$, as discussed earlier. Their embeddings $\epsilon(W) = \{\epsilon(w_1), \epsilon(w_2), ..., \epsilon(w_m)\}$ are then extracted. The embeddings are averaged, to find the average embedding of the section, as:

$$E(W) = \frac{\sum_{i=1}^{m} \epsilon(w_i)}{|W|} \quad (3)$$

	#CVs	#Sections	$\frac{\#Sections}{\#CVs}$
Train	1179	6085	5.2
Test	130	747	5.7

Table 1: Dataset statistics

We then extract the top M words, W', from the training-data vocabulary, based on the cosine similarity between the averaged embedding $E(W')$ and the vocabulary words W. Intuitively, these words act as the representative cluster of words, semantically most similar to the section content. The embeddings of these top M words in the vocabulary are then obtained as $\epsilon(W') = \{\epsilon(w_{1'}), \epsilon(w_{2'}), ..., \epsilon(w_{m'})\}$ and averaged in a manner similar to Equation 3, to obtain E(W'). Then, for each label $l_j \in L$, the cosine similarity of l_j and the averaged embedding $E(W')$ is calculated. The winner label is the one that shows up the maximum cosine similarity.

2.3 Split Section Approach

One of the main drawbacks of the schemes proposed is that they do not treat section headers and body-content separately. In practice, however, section headers can sometimes play a crucial role in determining the category that the section should belong to. This motivated us to propose other set of variants, in which section header and body-content are treated as two separate entities. The steps in the schemes proposed are carried independently on header and body-content. After lexical similarity with labels for both body and header are computed separately, the winner label is selected through voting. This idea lead to 5 more model variants such as SPLITPATH, SPLITWUP, SPLITCONCEPT, SPLITEMBEDDING, SPLITMULTIEMBEDDING.

We also implemented other variants such as: (a) averaging embeddings of words in the test data and then comparing the cosine similarity between the averaged resultant embedding with label embeddings (b) getting the top M words using WordNet and ConceptNet similarities. But these methods did not perform well, hence, results are not reported for these methods.

3 Experimental Setup

3.1 Dataset Creation and Preprocessing

Since, there is no publicly available standard CV dataset, we randomly pulled out 1309 number of CVs by requesting the recruitment division of a multinational organization (anonymized in this version). Since CVs can come up with different file formats (such as `pdf`, `html` *etc.*), we converted every CV to `docx` format using the `abiword` application[4], thereby preserving meta information about sections. The `docx` files are then processed using the `docx` package in `python` to separate out section headers and bodies for each CV. Textual noise in the form of non-Unicode characters and escape characters are then removed. Table 1 presents a detailed statistics about the number of CVs and number sections thus obtained.

3.2 Defining Labels

Our task intends to eventually help in analysis of CV by categorizing them, making it necessary for us to define a label-set that ensures decent coverage while maintaining a proper level of granularity. If the labels are too coarse or too fine, it will considerably increase the effort of analyzing the CVs and our task will be ineffective. We, thus, carefully chose 30 labels from the `Text Kernel`[5] platform, which provides a considerable coverage while balancing the granularity. The labels are shared in the supplementary material. In future, we plan to include important multi-word labels in our label set.

3.3 Test Data Annotation

To evaluate our methods against ground truth, we employed two software professionals (with acceptable working proficiency in English) to annotate the sections in the test data. The inter-annotator agreement between the annotators turns out to be 669 out of the 747 sections (89.56%), with 78 non-agreements. We manually inspect all the cases of non-agreement, and find that these are very similar. Some examples of such confusion pairs are *skills* vs. *interests*, *training* vs. *internship*, etc. In order to resolve, in a label preprocessing step, we randomly choose one of the non-agreeing two labels and assign the chosen label to the test instances, before we perform ground

[4]https://www.abisource.com
[5]https://www.textkernel.com

truth validation. The labels provided by the annotators are compared with the output generated by our system, to obtain precision, recall, accuracy and F-score measures.

4 Results and Insights

We present the results and insights obtained from the experiments in this section.

4.1 Results

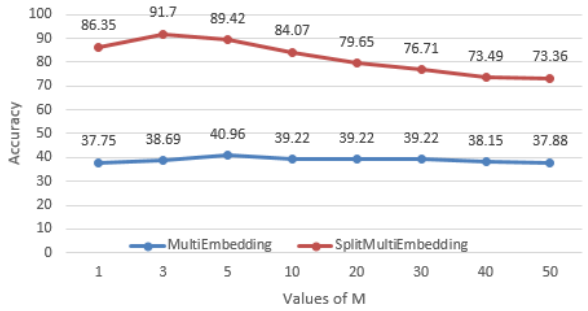

Figure 1: Variation of accuracy with M, the number of representative words chosen

From Table 2, we observe that the SPLITEMBEDDING method, which is the embodiment of Scheme 1 (given in Section 2.1) where $\sigma(w_i, l_j)$, the semantic similarity of word w_i with the label l_j, is computed using embedding, yields the highest precision across all the methods. However, SPLITMULTIEMBEDDING, a variant of Scheme 2 (given in Section 2.2) where the embeddings of section header and body are independently computed, and a weighted combination of the embeddings is used to retrieve the representative words of the section to compare with the embeddings of the labels, delivers the highest recall and F-score values, as well as, the highest overall accuracy. Thus, the SPLITMULTIEMBEDDING approach with $M = 3$ empirically turns out to be the most effective approach. Overall, 4 of the approaches deliver strong performances (F-score $>$ 80%): SPLITMULTIEMBEDDING, SPLITEMBEDDING, SPLITPATH and SPLITWUP.

Figure 1 shows the impact of varying M, on the system accuracy, for the MULTIEMBEDDING and SPLITMULTIEMBEDDING approaches. It is evident from the figure that for SPLITMULTIEMBEDDING, the most effective value is $M = 3$, while for SPLITEMBEDDING the value is $M = 5$. Beyond these values of M, too many words get chosen, which in turn confuses the system.

4.2 Error Analysis

We investigate the errors that our system makes, by comparing the section headers we obtain, with ground truth. Table 3 captures a random sample of the classifications made by our system. Note that, CVs that contain Personal sections (including name, email and other details inside the section), have always been classified with 100% accuracy. This also applies for CVs that have separate section headers for identification, such as Name, Email *etc*. On the other hand, for sections that are intuitively more complex, show some (meaningful) confusions across classes. For example, one would naturally assume Activities to have semantic overlaps with Skill, and similarly Work with Internship, and Project with Publications, among others. A few confusions are more intriguing, such as Project with Country (1 instance), and Education with City (1 instance). These confusions are rare although existent, showing the effectiveness of our system in general though there are perhaps some corner cases that can potentially be improved in the future.

5 Discussion

One aspect to note is that the approaches where the CV section header and body content are split, and the embeddings are subsequently combined in a weighted manner, outperform the approaches where the section header and body are given equal weightage. This conforms to the intuition that section headers bear a higher significance, compared to words that tend to appear in section bodies.

Further, we observe that the word embedding based approaches consistently and significantly outperform the WordNet and ConceptNet based approaches. While WordNet and ConceptNet are valuable lexical resources on their own, but clearly a predefined knowledge representation proves to be inadequate to capture the intricacies that CVs tend to present in real life. This highlights (a) the inherent challenge in dealing with the semi-structured and heterogeneous data that CVs present to computational systems, as well as (b) the importance of learning the lexical characteristics from the core application domain.

6 Conclusion

In this paper, we posed the restructuring of CVs as a section relabeling problem. We proposed a methodology to reassign a predefined label to each

Approach	Precision	Recall	F-Score	Accuracy
PATH	79.224	50.870	61.96	50.870
SPLITPATH	92.367	83.936	87.95	83.936
WUP	48.838	30.656	37.67	30.656
SPLITWUP	92.223	83.936	87.88	83.936
CONCEPT	58.241	43.507	49.81	43.507
SPLITCONCEPT	77.054	72.155	74.52	72.155
EMBEDDING	79.277	30.522	44.07	30.522
SPLITEMBEDDING	**95.029**	86.613	90.63	86.613
MULTIEMBEDDING ($M = 3$)	77.183	38.688	51.54	38.688
SPLITMULTIEMBEDDING ($M = 3$)	94.687	**91.700**	**93.17**	**91.700**

Table 2: Results for relabeling task for multiple approaches, numbers are shown in %

Ground Truth	Total	Correct	List of Confusions
PROJECT	110	99	Publication: 8, Training: 1, Activities: 1, Country: 1
EDUCATION	102	101	City: 1
ACTIVITIES	36	28	Skill: 3, Interest: 1, Work: 1, Hobby: 1, Publication: 1, Country: 1
PUBLICATION	30	26	Reference: 4
WORK	32	23	Inernship: 5, Reference: 2, Interest: 2
SKILL	87	84	Interest: 1, Education: 1, Objective: 1
PERSONAL	61	61	—
NAME	57	57	—
EMAIL	9	9	—

Table 3: Confusion matrix, showing some randomly chosen ground truth classes from actual CV section headers, and our system predictions in the form of <incorrect class: incorrect classification count of our system for that class>

section of given CVs, learning phrase embeddings from a pool of training CVs, and exploring several heuristics to compute the semantic relatedness between section headers, section contents and available labels. Our best heuristic achieves an F-score of 93.17% on a test dataset, with gold-standard labels obtained using manual annotation. Our system is useful in practical scenarios such as applicant management for recruitments, employee career management, and automated CV creation and maintenance for individuals.

References

Zhang Chuang, Wu Ming, Li Chun Guang, Xiao Bo, and Lin Zhi-qing. 2009. Resume parser: Semi-structured chinese document analysis. In *Computer Science and Information Engineering, 2009 WRI World Congress on*. IEEE, volume 5, pages 12–16.

Catherine Havasi, Robert Speer, and Jason Alonso. 2007. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent advances in natural language processing*. John Benjamins Philadelphia, PA, pages 27–29.

Sunil Kumar Kopparapu. 2010. Automatic extraction of usable information from unstructured resumes to aid search. In *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*. IEEE, volume 1, pages 99–103.

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.

Hugo Liu and Push Singh. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal* 22(4):211–226.

Sumit Maheshwari, Abhishek Sainani, and P Reddy. 2010. An approach to extract special skills to improve the performance of resume selection. *Databases in Networked Information Systems* pages 256–273.

Ujjal Marjit, Kumar Sharma, and Utpal Biswas. 2012. Discovering resume information using linked data.

International Journal of Web & Semantic Technology 3(2):51.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. 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.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, pages 45–50. http://is.muni.cz/publication/884893/en.

Amit Singh, Catherine Rose, Karthik Visweswariah, Vijil Chenthamarakshan, and Nandakishore Kambhatla. 2010. Prospect: a system for screening candidates for recruitment. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, pages 659–668.

Steve Spagnola and Carl Lagoze. 2011. Edge dependent pathway scoring for calculating semantic similarity in conceptnet. In *Proceedings of the Ninth International Conference on Computational Semantics*. Association for Computational Linguistics, pages 385–389.

Melanie Tosik, Carsten Lygteskov Hansen, Gerard Goossen, and Mihai Rotaru. 2015. Word embeddings vs word types for sequence labeling: the curious case of cv parsing. In *VS@ HLT-NAACL*. pages 123–128.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 133–138.

Kun Yu, Gang Guan, and Ming Zhou. 2005. Resume information extraction with cascaded hybrid model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 499–506.