# NLP methods for automatic candidate's CV segmentation

Maria Tikhonova
*Doctoral School of Computer Science*
*Sberbank,*
*Higher School of Economics*
Moscow, Russia
m_tikhonova94@mail.ru

Anastasia Gavrishchuk
*Faculty of Communications, Media, and Design*
*Sberbank,*
*Higher School of Economics*
Moscow, Russia
angavrishuk@gmail.com

***Abstract*—The problem of CV (or resume) segmentation and automatic extraction becomes increasingly relevant nowadays as long as it could simplify candidate selection process. The paper proposes a new method of automatic CV segmentation and parsing. The described algorithm is based on Natural Language Processing and Machine Learning methods. The proposed procedure allows to extract information related to the candidates' work experience and education from their CVs which come in pdf or docx format. In particular, CV segmentation into 3 blocks (*Basic Information, Education* and *Work Experience*) is performed.**

***Keywords* — NLP methods, Machine Learning, topic modeling, Natural Language Processing, text segmentation, word embedding**

## I. Introduction

The problem of automatic CV (or resume) information extraction and parsing is actual, especially in big companies where the flow of new candidates' is high and manual CV processing is time-consuming and demands a lot of human resources. In this context the task of automatic CV parsing in order to extract information about the candidate's work experience, education, skills, additional education and general information arises. The extracted information could further be used to simplify the work of the recruiter as long as it reduces resume processing time.

The article proposes an algorithm for automatic CV parsing that is extraction information about work experience, education and basic info, which is based on *Natural Language Processing methods*. For each text line in a CV it is predicted whether it contains information about work experience or education and then CV segmentation into 3 blocks (*basic Information, education* and *work experience*) is performed.

The second part of the paper contains a detailed description of the proposed procedure. The third part describes the algorithm's application for real CVs and presents the results of the conducted experiments. The procedure was tested on candidates' CVs applying in Sberbank for different vacancies. In addition, in the fourth part of the paper error analysis is conducted and ways of the algorithm's improvement are proposed.

It is worth noticing, that there exists a variety of standard CV forms (for example, a CV automatically created by *HeadHunter* (hh.ru) — one of the biggest Russian job search sites). As long as such CVs are subject to the strict pattern, they could be easily segmented and processed via rule-based approach and regular expressions. However, there still exists a large percentage of CVs which are not standardized. Forms, fonds, general layout and the way of putting information in them could be quite unconventional and, therefore, rule-based approach fails for such CVs. In our research we focus mostly on the latter nonstandard CV segmentation.

## II. Algorithm's description

For each line the proposed algorithm performs three class classification (basic information, education or work experience) and, therefore, a CV could be separated into segments, where each segment corresponds to adjacent lines of the same type (see fig. 1).

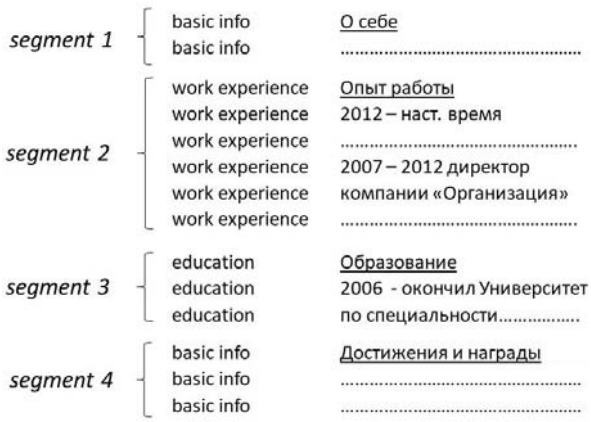


Figure 1: Way of CV segmentation.

The CV segmentation procedure consists of 7 consecutive steps:

**Step 1.** CV prepossessing and transformation;
**Step 2.** Word embeddings construction;
**Step 3.** Computation of tf-idf index;
**Step 4.** Text field embeddings construction;
**Step 5.** Specific features extraction;
**Step 6.** Text Line classification;

**Step 7.** Final CV segmentation.
Below each step is described in more detail.

### A. CV prepossessing and transformation

On the first step each CV is read from a pdf (or docx) file as plain text written line by line. The text is further transformed into the format suitable for machine learning algorithms. Namely, the algorithm runs through the text in a sliding window shifting one line down on each iteration and takes a text part which:

1) starts from the current line;
2) contains 25 or more words;
3) its end matches the end of some line in the text.

Thus, for every CV we obtain a number *text windows* (or text parts) which are in one-to-one correspondence with CV text lines. Therefore, instead of classifying text lines, whose length can vary significantly, it is sufficient to classify more standardized text windows.

On the following steps text windows are further transformed into numerical format via NLP-methods suitable for machine learning algorithms.

### B. Word embeddings construction

On the second step words which are contained in CVs collection are embedded into the linear space $\mathbb{R}_n$. Thus, for every word we obtain an $n$-dimensional vector. Moreover, there exists a number of word embedding algorithms which construct word embeddings preserving syntactic and semantic word properties. Among the most popular are Word2Vec [1], [2], FastText [3], [4] and GloVe [5]. In addition, pretrained word embeddings, built of large collections of documents, are available for free use in many different languages including Russian.

### C. Computation of tf-idf index

On the third step for each word in every text window its *tf-idf* index is computed. *Tf-idf (term frequency-inverse document frequency)* is a statistic which represents the importance of a word in a document of the collection. It is proportional to the frequency of a word in a document and inversely proportional its frequency in the collection of documents. Thus, tf-idf discriminates rare words which are frequent in a specific document. That is why in information retrieval it is often used as a weighting factor.

$tf$ of a word $t$ in a document $d$ is defined as:

$$tf(t,d) = \frac{n_t}{\sum\limits_{k \in d} n_k} \tag{1}$$

$n_t$ — a number of times a word $t$ appears in a document $d$.

$idf$ of a word $t$ in the collection of documents $D$ is defined as:

$$idf(t,D) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|} \tag{2}$$

where $|D|$ denotes a number of elements in a set $D$.

Finally, *tf-idf* of a word $t$ in document $d$ which belongs to the collection of documents $D$ is computed as the product of $tf(t,d)$ and $idf(t,d,D)$:

$$tf - idf(t,d,D) = tf(t,d) \cdot idf(t,d,D). \tag{3}$$

### D. Text field embedding construction

Next, each CV part obtained on the first step is embedded into an $n$-dimensional vector. For this purpose, vector representations of the words present in the CV are summed with their tf-idf weights:

$$v(d) = \sum_{w \in d} tf - idf(w,d) \cdot v(w) \tag{4}$$

$d, w$ — a text field in the CV and a word in d, respectively,
$v(d)$, $v(w)$ — embeddings for d and w, respectively,
$tf$- $idf(w,d)$ — tf-idf index of $w$ in $d$.

Thus, each text window is transformed into an $n$-dimensional feature vector which could be used as an input for a machine learning algorithm.

### E. Specific features extraction

On top of that feature space is expanded with specific features, which allow to increase classification results:

1) Part-of-Speech counters — the number of every part of speech in the text window (noun counter, verb counter, adjective counter, etc.). In total 11 new features corresponding to different parts of speech are created.

2) Counters of the "specific suffixes" — the number of words in the text window which contain suffixes specific for the work experience and education description In the current version of the algorithm for suffix features are computed which correspond to Russian suffixes -ик, -ирование, -истика and -ость.

### F. Text Line classification

After the data are embedded into the feature space, the problem of CV segmentation is regarded as two binary classification tasks:

1) *work experience prediction* — for each text window it is predicted whether it belongs to work experiment segment or not.

2) *education prediction* — whether a text window belongs to the education segment

On top of the classification results some rule-based heuristics are added:

1) Predictions are smoothed in a sliding window in order to get rid of outliers. Sliding window size is a parameter which could be choicen on the validation set and which normally lies between 5 and 10 and.

2) Small segments of the size less than 5 text windows are deleted.

For instance, in fig. 2 for the prediction (left columns) with 4 initial segments we obtain adjusted prediction (right column) with only 2 segments.

3) When small segments are removed bounds of the remaining segments are adjusted. For each bound it is verified

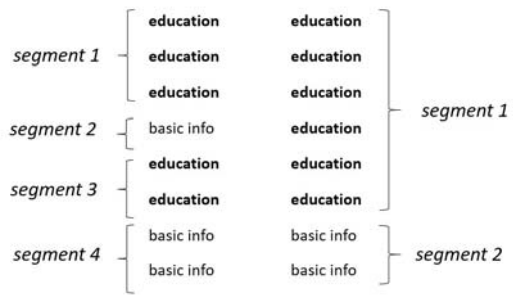


Figure 2: Example of small segments deletion.

whether any line in a *gap-window* contains words which indicate the beginning of the new segment. For example, words such as *work experience, skills, education, additional education* are regarded as markers of the education block. Gap-window size is another parameter which could be found via grid search on a validation set. Normally, it s chosen between 5 and 12. If any of the words is found in text window $w_i$, the segment border is moved to the text window $w_i$.

### G. Final classification

Finally, when the heuristics listed above are applied to the work experience and the education prediction, the two binary predictions are merged and the final segmentation is obtained according to the following rule:

$$pred_{final} = \begin{cases} \text{education} & pred_{edu} = 1 \\ \text{work experience} & (pred_{edu} = 0) \wedge \\ & (pred_{we} = 1) \\ \text{basic info} & \text{otherwise.} \end{cases} \quad (5)$$

$pred_{edu}$ — prediction of the education clasificator,
$pred_{we}$ — prediction of the work experience classificator,
$pred_{final}$ — final segmentation result.

We also experimented with the equation where the preference was given to work experience prediction:

$$pred_{final} = \begin{cases} \text{work experience} & pred_{we} = 1 \\ \text{education} & (pred_{we} = 0) \wedge \\ & (pred_{edu} = 1) \\ \text{basic info} & \text{otherwise.} \end{cases} \quad (6)$$

However, the latter strategy showed worse results on validation set and, therefore, equation 5 was chosen.

## III. Experiments

### A. Data

*1) Training set:* as long as we did not possess a collection of labelled nonstandard CVs large enough (or the resources to label enough CVs) to train the classifiers, we used data available on *HeadHunter* — the major Russian job search site. We downloaded 5000 anonymized CVs from HeadHunter already segmented into 13 categories, which was actually surplus for our task. We, therefore, merged the excessive categories and left only "basic info" , "work experience" and "education" .

*2) Test set:* we selected 50 random CVs from the pool of candidates applying for Sberbank and manually labelled (or sectioned) them into 3 categories by assigning each string to one of the tree categories identical to the ones in the training set.

### B. Evaluation

For the evaluation we used *Jaccard index* [7], also known as *Intersection over Union*. It measures similarity between two sets and is defined as the size of the intersection divided by the size of the union of the sets:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \quad (7)$$

and if $|X| = |Y| = 0$, then $J(X,Y) = 1$.

Jaccard index varies between 0 and 1. The bigger the Jaccard index is, the more is the sets intersect.

We also experimented with the metric known as WindowDiff [6] which is commonly used for the segmentation task. However, as long as in a typical CV the number of segments was quite limited, Jaccard index turned out to be more representative. Thus, we decided to use it as a main measure of quality.

### C. Model description

All the experiments were implemented on *Python*. To obtain plain text from a pdf or docx document Python libraries *textract* and *python-docx* were employed. We used 25+ window size and trained two *catboost classifiers* (https://yandex.ru/dev/catboost/) independently for classifying work experience and education. Parameters for the final models and heauristics were selected via grid search and are listed in Table I.

Table I: Model parameters

| Parameter name | Work Experience | Education |
|---|---|---|
| n_estimators (catboost) | 500 | 500 |
| learning_rate (catboost) | 0.1 | 0.09 |
| depth (catboost) | 9 | 8 |
| l2_leaf_reg (catboost) | 10 | 10 |
| smoothing window size | 10 | 6 |
| gap-window size | 5 | 12 |

## IV. Results

Jaccard index was computed independently for work experience and education on test set. With the parameters listed above the following results were obtained:

$Jaccard_{work\ experience} = 0.942$,
$Jaccard_{education} = 0.806$.

In addition, the number of *well-segmented* CVs was measured.

**Definition.** A CV $C$ is call *well-segmented* $\iff \mathrm{J}_{work\_exp}(C) + J_{edu}(C) > 1.7$.

| Model prediction | True Class | Text |
|---|---|---|
| basic info | basic info | Иванова Анна Ивановна |
| basic info | basic info | +70123456789 — предпочитаемый способ связи |
| basic info | basic info | 1111@mail.ru |
| basic info | basic info | Город: Тюмень |
| basic info | basic info | Желаемая должность |
| basic info | basic info | Бухгалтер-экономист |
| basic info | basic info | Занятость: стажировка, частичная занятость, полная занятость |
| basic info | basic info | График работы: гибкий график, полный день |
| work experience | work experience | **Опыт работы —9 лет 4 месяца** |
| work experience | work experience | **Апрель 2009 —настоящее время** |
| work experience | work experience | **Сбербанк России** |
| work experience | work experience | **ведущий специалист** |
| work experience | work experience | **Начальник СОФЛ. За время моего декретного отпуска ставку** |
| work experience | work experience | **сократили. Сейчас я - ведущий специалист** |
| work experience | work experience | **Должностные обязанности: руководить персоналом. Чтобы** |
| work experience | work experience | **филиал бесперебойно работал и приносил прибыль. Достижения - Превышали все** |
| work experience | work experience | **показатели по продажам продуктов Сбербанка. Со всеми поставленными задачами** |
| work experience | work experience | **справлялась во время.** |
| education | education | Образование |
| education | education | Высшее |
| education | education | 2008 Курганская Государственная Сельскохозяйственная Академия им. |
| education | education | ТС Мальцева |
| basic info | education | Финансы и кредит, Налоги и налогообложение. |
| basic info | basic info | Дополнительная информация |
| basic info | basic info | Русский — родной |
| basic info | basic info | Английский — базовые знания |
| basic info | basic info | Обо мне |
| basic info | basic info | Очень ответственно отношусь к работе и поставленной задаче, коммуникабельна |
| basic info | basic info | целеустремленная, работоспособность, легкая обучаемость, организаторские |
| basic info | basic info | способности. |

Figure 3: Example of a well-segmented CV.

| Model prediction | True Class | Text |
|---|---|---|
| work experience | work experience | **Контролер-кассир Сектора по работе с физическими лицами июль 2007 - март 2008** |
| work experience | work experience | **обслуживание физических лиц по совершению операций;** |
| work experience | work experience | **выполнение плана продаж услуг клиентам Банка.** |
| work experience | basic info | Личные достижения |
| work experience | basic info | награждена благодарственной грамотой Банка в честь 167-летия Банка (2010г.); |
| work experience | basic info | награждена дипломом о внесении вклада в работу Поволжского Банка (2012г.); |
| work experience | basic info | получила благодарность за высокие достижения в инновационной деятельности Банка |
| work experience | basic info | награждена благодарственной грамотой Банка в честь 175-летия Банка (2016г.); |
| work experience | basic info | получила благодарность за успешную реализацию проекта «Новый формат привлечения |
| work experience | basic info | пенсионеров» (2017г.). |

Figure 4: Wrong acknowledgments classification.

The number of well-segmented CVs equaled 41 (out of 50), which covers 82% of the test set. Moreover, out of these 41 CVs 21 were classified correctly without errors (42% of the test collection). An example of a well-segmented CV with only minor errors is given in fig. 3.

## V. Error analysis

The major part of the errors on the test set was caused by the luck of variety in the training data as long as the whole collection was of the standard HeadHunter type.

For instance, the Education part on HeadHunter contains only information about higher education without and courses or additional education. Thus, trained on such data classifier made errors classifying mention of courses in a CV, such as Coursera (https://www.coursera.org).

Similar problem was with classifying acknowledgments which was often incorrectly labelled as work experience ((see fig. 4).

Thus, our current goal is to augment the data by enriching it which untypical nonstandard CVs, which do not fall under any pattern.

## VI. Conclusion

In the paper a new method of automatic CV parsing and segmentation was described which allows to extract information about the candidate's work experience and education from a text document in pdf or docx format.

In the future we plan to increase the algorithm's quality by enriching training data with nonstandard CV examples. Moreover, we also plan to expand the algorithm for the extraction of other segments such as *skills, additional education, etc*.

## REFERENCES


[1] Mikolov, Tomas, et al. “Efficient estimation of word representations in vector space.”arXiv preprint arXiv:1301.3781 (2013).

[2] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. “Linguistic regularities in continuous space word representations.”Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013.

[3] Bojanowski, Piotr, et al. “Enriching word vectors with subword information.”Transactions of the Association for Computational Linguistics 5 (2017): 135-146.

[4] Joulin, Armand, et al. “Bag of tricks for efficient text classification.”arXiv preprint arXiv:1607.01759 (2016).

[5] Pennington, Jeffrey, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation.”Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[6] Pevzner, Lev, and Marti A. Hearst. “A critique and improvement of an evaluation metric for text segmentation.”Computational Linguistics 28.1 (2002): 19-36.

[7] Jaccard, Paul. “Etude comparative de la distribution florale dans une portion des Alpes et des Jura.”Bull Soc Vaudoise Sci Nat 37 (1901): 547-579.