Automated Analysis of Job Requirements for Computer Scientists in Online Job Advertisements

Joscha Grüger[1,2][a] and Georg J. Schneider[1][b]

[1] *Computer Science Department, Trier University of Applied Sciences, Main Campus, Trier, Germany*
[2] *University of Trier, Department of Business Information Systems II, 54286 Trier, Germany*

Keywords: Data Analysis, Web Mining, Natural Language Processing, Information Retrieval, Machine Learning, Job Ads, Skills.

Abstract: The paper presents a concept and a system for the automatic identification of skills in German-language job advertisements. The identification process is divided into Data Acquisition, Language Detection, Section Classification and Skill Recognition. Online job exchanges served as the data source. For identification of the part of a job advertisement containing the requirements, different machine-learning approaches were compared. Skills were extracted based on a POS-template. For classification of the found skills into predefined skill classes, different similarity measures were compared. The identification of the part of a job advertisement containing the requirements works with the pre-trained LinearSVC model for 100% of the tested job advertisements. Extracting skills is difficult because skills can be written in different ways in the German language – especially since the language allows ad-hoc creation of compound. For extraction of skills, POS templates were used. This approach worked for 87.33% of the skills. The combination of a fasttext model and Levenshtein distance achieved a correct assignment of skills to skill classes for 75.33% of the recognized skills. The results show that extracting required skills from German-language job ads is complex.

1 INTRODUCTION

The labor market for IT specialists is very complicated due to the always evolving and changing IT environment. New programming languages, frameworks and development concepts appear on a yearly basis. Some of these technologies and concepts are very trendy only for a certain period, some manifest and play an important role for a long time. For a company it is vital to find IT specialists that are competent with the strategic technologies used in their enterprise. In a study by Bitkom in the year 2015, seven out of ten ICT companies (70 percent) state that there is currently a shortage of IT specialists. In 2014, only 60 percent of those surveyed said that there was a shortage (Bitkom, 2016). 38.7 percent of the IT vacancies forecasted for the top 1000 companies in Germany in 2015 are difficult to fill due to a lack of a fitting qualification, and 6.1 percent of the vacancies will remain vacant due to a lack of suitable candidates (Weitzel et al., 2015). Hence it is vital for a national economy to educate students that have not only an excellent qualification in the theoretical foundations of Computer Science but also up-to-date skills in current technologies for this urgent need. Likewise, being competent with technologies in demand, it helps the alumni to easily find a job.

Universities of Applied Sciences in Germany usually target praxis-oriented topics and they have traditionally a close link to the professional field. Therefore lab courses, student projects or applied research projects usually reflect the current trends in the IT to complete the theoretical foundations with practical experiences using modern IT systems. Hence the students can easily find a job with the competencies acquired during their education. Universities try to solve the problem of staying up-to-date with their education by regularly reviewing their curricula. They observe developments in the industry to offer either specialized courses that complement the sound theoretical foundation of their study program or to change programming languages, frameworks or systems for the lab courses or projects. Students usually embrace these changes and also gladly choose new course offerings to get the best possible chances on the labor market. For both students and universities, the question is, what should be taught on the practical side and

[a] https://orcid.org/0000-0001-7538-1248
[b] https://orcid.org/0000-0001-7194-2394

Grüger, J. and Schneider, G.
Automated Analysis of Job Requirements for Computer Scientists in Online Job Advertisements.
DOI: 10.5220/0008068202260233
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 226-233
ISBN: 978-989-758-386-5

what should students learn to best meet the requirements of the industry? One possibility would be to manually review the job advertisements in different newspapers and online platforms. However, this is a tedious and time-consuming task. Our research hypothesis is that this task can be automated in a way that the findings are at least nearly as good as a human. German, however, is a challenge compared to English, where similar approaches already exist, as German has a rich morphology, umlauts, four cases and much fewer corpora for training.

This paper describes an approach for an automated survey of current job requirements for computer scientists on the German labor market. Obviously, this survey has to be carried through regularly to identify trends and manifesting technologies.

The suggested process to realize the automated survey is based on technologies of Natural Language Processing and Machine Learning. The results of the analysis can be used, for example, to check the content of university curricula or to identify new technological trends at an early stage. In addition, the procedure can be used to implement a skill-based job search. The next section discusses the current state of research and various papers regarding the extraction of skills from job ads. Section 3 introduces our approach to job ad analysis. Both the processing steps and the evaluation of each step are described in detail. Finally, Section 4 concludes the results and discusses future work.

2 FOUNDATIONS AND RELATED WORK

Some studies conclude that education does not comply with the requirements of employers in the IT sector. Kwon Lee and Han (2008) for example concluded for the US labor market that most universities attach great importance to hardware and operating systems, although the employers surveyed are rarely interested in these skills. They see for example deficits in the teaching of skills in the economic and social category. Yongbeom et al. (2006) also speak of a skill gap between employers' requirements and universities' IT curricula. Among other things, the lack of project management, Enterprise-Resource-Planning (ERP) and information security modules in the curricula is criticized. Scott et al. (2002) criticize poor database knowledge and the lack of skills in CASE/-modeling tools and Business Process Reengineering (BPR) techniques among graduates. Students also lacked skills in XML and iterative development.

There are a number of reasons for the skill gap: one is the rapid technological change, another is the discordance between the content of the curricula of the universities and the required competencies of the industry (Scott et al., 2002; Milton, 2000). In addition, too long revision cycles of curricula relative to the speed of technology change and a lack of knowledge at universities about new and upcoming technology are cited as reasons for the gaps (Lee et al., 2002).

In order to keep curricula up-to-date, universities need to know which competencies are currently and in the long term required by the industry. For identifying the skills required various approaches exist. Prabhakar et al. (2005) researched online job advertisements for computer scientists in the US in 2005 with regard to the changing demand for skills over the year. For this purpose, they examined the job advertisement to see whether it contained one of 59 keywords or not. The approach for identifying requested skills in the IT sector of Gallagher et al. (2010) is interview-based. His team interviewed 104 senior IT managers. The questions were very general, e.g. it was asked whether programming skills were required, and not for concrete programming languages like Java. Sibarani et al. (2017) developed an ontology-guided job market demand analysis process. Their method is based on the self-defined SARO ontology and a defined set of skills. Using the predefined skills and ontology, they perform a named-entity tagging. The identified skills are linked by a co-word analysis. Litecky et al. (2010) used web and text mining techniques for retrieving and analysing a data set of 244.460 job ads. They scraped the data from online job exchanges to extract titles and requirements based on predefined keywords. Wowczko (2015) took a different approach. They analysed descriptions of vacancies and reduced the words used in the descriptions until only significant words remained. Custom word lists, stemming, removing stopwords, removing numbers, stripping whitespaces, etc. were used to clean up the data.

The problem with all these approaches except Wowczko (2015) and Gallagher et al. (2010) is that they are based on fixed keyword lists. Thus, only abilities contained in the lists are recognized. New technologies or skills described in any other way cannot be recognized. Abbreviations like *Active Directory* and *AD* are assigned to different classes or remain unrecognized. Additionally, some processes were performed manually and are hence rather time-consuming and are only carried out periodically. Wowczko (2015) also finds false positives like *strong*, *excellent* and *can*. Consequently, an automated procedure that monitors job advertisements permanently would simplify this procedure enormously. Thus the approaches also search in areas of the job advertisement where no requirements are described (e.g. in the company description).

This paper describes a concept and a resulting automated procedure to search for competencies in all job advertisements. The approach also recognizes unknown competencies and maps them to the same class if they are semantically similar or equal. For the extraction of the skills, only the area in which the requirements for the applicant are formulated is used.

3 CONCEPT

The focus of the approach is on job advertisements in German. The target group of the analyzed job advertisements are computer scientists. Online job exchanges such as monster.de and stepstone.de were used as data sources. The process for identifying skills is divided into four steps (see figure 1): Data Acquisition, Language Detection, Section Classification and Skill Recognition.

3.1 Data Acquisition

A web crawler based on the Scrapy framework was developed for data retrieval. The web crawler searches German online job exchanges for jobs in the IT sector. The online job exchanges monster.de, stepstone.de and stellenanzeigen.de were used as data sources. The crawler extracts the HTML job ads found, as well as metadata such as company name, job title, and work location. The defined process works on the HTML files of the job advertisements. The metadata could be used for extensions such as geographic analysis or job title analysis.

3.2 Language Detection

The result of the data acquisition are all job ads, directed at computer scientists of the online job exchanges mentioned. Seven percent of the extracted job advertisements are written in English, 93% of the advertisements are written in German. As described, the approach is aimed at ads in German, the English-language job advertisements must be filtered out.

In order to be able to filter out non-German advertisements a language recognition according to Shuyo was implemented (Shuyo, 2010). The basic assumption of the approach is that certain n-grams occur in different frequency in different languages (see table 1). Based on a quantitative corpus analysis using Wikipedia pages, all mono-, bi- and trigrams of the languages to be detected were counted and the probability with which each n-gram occurs in the respective language was calculated. Shuyo provides the results of the quantitative analysis in so-called profile files.

Table 1: Comparison of the probability with which the n-grams occur in the languages German and English (based on Shuyos profile files.

n-gram	German	English
D	0.00648	0.00253
ie	0.01158	0.00259
hum	0.00013	0.00029

To recognize the language, the text to be classified is fragmented into mono-, bi- and trigrams. The algorithm randomly selects individual n-grams and calculates the probability with which these occur in the languages tested (Shuyo, 2010).

For evaluation, 741 job advertisements were classified by language. 8.77% of the job ads tested were in English, 91.23% in German. The algorithm achieved an accuracy of 100% and examined a maximum of 15 n-grams per text.

3.3 Section Classification

After all English-language job adverts have been filtered out, the position of the requirements within the job advertisement must be identified. Position means in which HTML list element the requirements are listed. For this purpose classification algorithms of machine learning were trained and compared.

For training 150 job advertisements were segmented. To this end, all HTML list elements with more than 20 characters contained in the job advertisements were assigned to a category. The categories are *Requirements*, *Offer*, *Tasks* and *uncategorized*. Requirements describes the employer's requirements for the applicant. Elements classified as tasks contain the tasks to be performed on the position. Offer-sections contain information about the offers and benefits of the company. The models were trained based on the segmented data of 50 job advertisements. The test corpus includes 100 segmented job advertisements.

The scikit-learn[1] implementations of RandomForestClassifier, LinearSVC (Linear Support Vector Classification), MultinomialNB (Naive Bayes classifier for multinomial models), LogisticRegression, DecisionTreeClassifier and BernouliNB (Naive Bayes classifier for multivariate Bernoulli models) were trained and compared in the scikit-learn standard configuration. For tokenization, a word tokenizer was used. Features were represented by a count matrix. The result of the count matrix was transformed into normalized tf (term frequency) representation, containing 3349 features. To compare the algorithms, each algorithm was trained 5 times on 50 segmented job ads

[1] Version: 0.20.3 (scikit-learn.org)

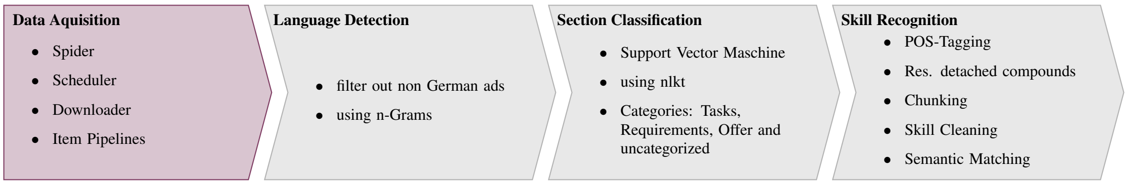

Figure 1: System architecture.

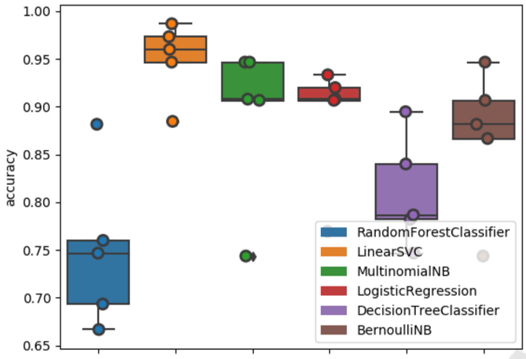

Figure 2: Box plot of 5 tests per algorithm. LinearSVC reached an accuracy of 0.950291.

Table 2: Comparing accuracy, precision and recall. The features were tokenized by words and the frequency measured by TF.

algorithm	accuracy	precision	recall
RandomForestCl.	0.7496	0.7215	0.5933
LinearSVC	0.9502	0.9416	0.9434
MultinomialNB	0.8902	0.8804	0.8103
LogisticRegr.	0.8874	0.8754	0.8043
DecisionTreeCl.	0.8100	0.8115	0.8249
BernoulliNB	0.8690	0.8185	0.7664

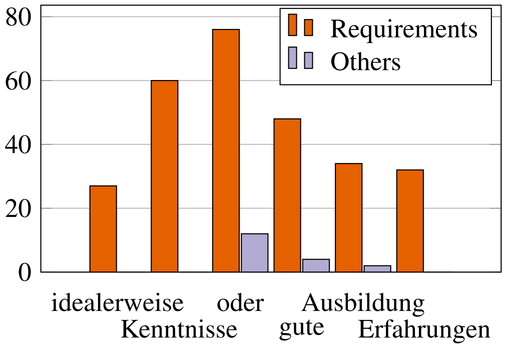

Figure 3: Frequency with which the features occur in requirement sections and in other areas (Tasks, Offer, uncategorized, based on 50 job ads).

Table 3: Comparing algorithms by accuracy (acc), precision (pre) and recall (rec) for different settings. For stemming NLTK (Natural Language Toolkit) German Snowball is used. The stop word list contains 621 German stop words. Pruning means removing tokens that are included in most job ads or in almost no job ads (< 0.5% or > 99.5%).

		LinSVC	Mult.NB	LogReg
TF-IDF	acc	0.9502	0.8902	0.8874
	pre	0.9416	0.8804	0.8754
	rec	0.9434	0.8103	0.8043
TF-IDF, prune < 0.5 % & > 99.5 %	acc	0.9368	0.8899	0.9421
	pre	0.9321	0.9543	0.9247
	rec	0.9308	0.9274	0.8163
TF-IDF, stop-words	acc	0.9372	0.9032	0.8821
	pre	0.9471	0.8875	0.8799
	rec	0.8960	0.8249	0.7963
TF-IDF, stop-words, stem	acc	0.9449	0.9137	0.9006
	pre	0.9512	0.9426	0.8895
	rec	0.9116	0.8424	0.8193
TF-IDF, stop-words, stem, prune	acc	0.9393	0.9000	0.9052
	pre	0.9245	0.8724	0.9248
	rec	0.9283	0.8205	0.8312
TF-IDF, stop-words, stem, n-gram(1,2)	acc	0.9163	0.8902	0.8240
	pre	0.9438	0.8820	0.8439
	rec	0.8436	0.8103	0.7227

and tested with 100 job ads. Rated by accuracy LinearSVC and MultinomialNB were the best algorithms. LinearSVC reached an average accuracy of 0.950291 and MultinomialNB 0.890297 (see figure 2 and table 2).

The results are achieved because skill/requirements sections contain words like `Kenntnisse` (eng. knowledge), `idealerweise` (eng. ideally) and `gute` (eng. good) in high frequency and almost exclusively (figure 3). For optimization, the classification algorithms were also tested with stemmed features, on a feature set with and without stop words, using n-grams, using TF-IDF (inverse document frequency) and with a pruned feature set. Table 3 shows that LinearSVC using TF-IDF, without further optimization, gives the best results in terms of skill section recognition.

The confusion matrix in table 4 shows that all requirement sections were correctly recognized. In addition to requirement sections, other areas were also classified as requirement sections. This leads to a precision of 0.826 for the Requirements class. For the further recognition of the skills, this would mean that skills would also be searched in e.g. offer sections. Recall and precision of the requirements class also

Table 4: Confusion matrix of prediction with LinearSVC and TF-IDF. The results show that texts of the classes Unclassified, Task and Offer were also classified as Requirements.

		Predicted				
	n = 266	Unc.	Offer	Req.	Task	recall
Actual	Unc.	14	2	1	0	0.823
	Offer	2	57	16	11	0.662
	Req.	0	0	100	0	1
	Task	1	11	4	47	0.746
	prec.	0.823	0.814	0.826	0.81	

Table 5: Confusion matrix of prediction with LinearSVC and TF-IDF under the premise that there is only one section containing requirements per job ad. The Requirement Section is the area for which the highest probability of containing requirements has been calculated.

		Predicted				
	n = 266	Unc.	Offer	Req.	Task	recall
Actual	Unc.	14	3	0	0	0.823
	Offer	2	76	0	8	0.883
	Req.	0	0	100	0	1
	Task	1	9	0	53	0.841
	prec.	0.823	0.864	1	0.869	

show that in some job advertisements several areas were classified as requirement sections. Therefore the premise was defined that in every online job advertisement exactly one requirement section exists. Based on the trained model, $P(x)$ calculates the probability with which text x is a requirement section. A requirement section r_i of a job advertisement i consisting of a set of section texts S_i is defined as:

$$r_i = max(\{P(s)|s \in S_i\}) \qquad (1)$$

Table 5 shows the result including the premise. No false positives are assigned to the requirements class and the recognition of the other classes also improves.

3.4 Skill Detection

After the requirement sections have been identified, the skills they contain must be extracted. A POS tagging was performed for this purpose. For the tagging the nltk ClassifierBasedTagger was trained with the Tiger Corpus of the Institute for Machine Language Processing of the University of Stuttgart. The Tiger Corpus consists of 90,000 tagged tokens and 50,000 sentences (IMS, 2018). The analysis of POS tagged skills shows that they can be included in requirement descriptions in a very heterogeneous form. Skills can include nouns, foreign words, compound words, detached compounds, combinations with cardinals, etc. Examples for skills in job ads are `Hochschulstudium`, `ISO 27001`, `gute Deutsch- und Englischkenntnisse` and `Erfahrung in der Programmierung mit Java`.

A multi-stage process was defined to identify the skills. First, detached compounds like German and English skills are resolved. In the next step, the skills get chunked by POS templates. False positives are then filtered out via a blacklist. In order to be able to count semantically similar skills in a class, they are assigned to a class via a semantic matching.

3.4.1 Resolve Detached Compounds

In German words can be concatenated to mean the same as the sum of two words (e.g. eng: administrator password – de: `Administratorpasswort`). By a conjunction, several compositions with the same end or beginning can be shortened without losing their meaning (e.g. `Clientadministrator und Serveradministrator` can be written as `Client- und Serveradministrator`). In order to understand the connections between the conjunctions, they must be resolved.

For resolving detached compounds like shown in the listing a compound resolver module was developed. The *compoundResolver*-module searches for tokens that are classified as *TRUNC* (Truncate) by the POS tagger and end with a hyphen punctuation mark. If the following token is classified as *KON* (Conjunction) and the following as *NN* (regular nominal) than the regular nominal is split into syllables using the *pyphen*[2] library. Examples of resolving detached compounds and the POS-tagging are:

```
Deutsch- und Englischkenntnisse
-> Deutschkenntnisse und Englischkenntnisse

Hochschul- oder Universitätsabschluss
-> Hochschulabschluss oder Universitätsabschluss
```

By merging the syllables, all possible combinations were generated while keeping the order of the syllables. Based on a dictionary of German nominals, each syllable is tested. The shortest combination contained in the Dictionary is identified as the first part of the compound. The rest of the word is merged with the truncated part of the compound. The new String replaces the truncated token. An example of the generated syllable combinations and the result after resolving is:

```
['Eng', 'Englisch', 'Englischkennt',
'Englischkenntnis', 'Englischkenntnisse']
```

[2] pyphen.org

```
[('Deutschkenntnisse', 'NN'), ('und', 'KON'),
('Englischkenntnisse', 'NN')]
```

For regular nouns that are not compounds (like in Wirtschafts- (Informatik)) the algorithm has been extended. By checking if just the whole word matches with an entry of the dictionary the token can be joined with the truncated token. An example of resolving Wirtschafts- (Informatik) is:

```
[('Wirtschafts-', 'TRUNC'), (')', '$('),
('Informatik', 'NN')]

[('Wirtschaftsinformatik', 'NN'), (')', '$('),
('Informatik', 'NN')]
```

3.4.2 Skill Chunking

Job requirements are specified in job advertisements in various forms. To extract the requirements, the text is chunked using POS templates and regular expressions. Four structures were identified for matching skills. Each structure is shown with examples and the appropriate expression for the chunking with *nltk RegexParser*.

Sequence of nouns, cardinal numbers, proper names or/and foreign words like: Deutsch, SAP R3, HTML Kenntnisse, Windows 10. The code shown indicates that the search is for single or consecutive occurrences of foreign words, normal nouns, proper names, or cardinal numbers.

```
<NN|NE|CARD|FM>+>
```

These sequences can be separated by hyphens, such as FH-Studium, ERP-Systeme, Microsoft-Zertifizierung, IT-System-Kaufmann, SAP IS-A.

```
<NN|NE|CARD|FM|TRUNC>+
```

Within a skill description, articles, prepositions, or prepositions with articles can be included: Kenntnisse [der] Informatik, Erfahrung [im] Projektmanagement, Erfahrung [in der] Arbeit [mit] Eclipse.

```
<NN|NE|CARD|FM|TRUNC>+<ART|APPR|APPRART>+
<NN|NE|CARD|FM|TRUNC>+
```

In addition, the description of a skill can contain attributive adjectives, adverbial or predicative adjectives: Kenntnisse [in der] technischen Informatik, Erfahrung [in der] objektorientierten Programmierung.

```
<NN|NE|CARD|FM|TRUNC>+<ART|APPR|APPRART>+
<ADJAD|ADJA>*<NN|NE|CARD|FM|TRUNC>+
```

These regular expressions were testet on a corpus of 60 job ads. 87.33% of the skills mentioned in the job ads could be recognized correctly. 12.67% of the skills could not be extracted. In addition, 21.57% of the extracted tokens are false positives.

3.4.3 Skill Cleaning

After the *Skill Chunking* the result contains 21.57% false positives like Wort or Fach. These are not job requirements and it is necessary to remove them from the results. For this purpose, the results are cleaned via a filter. The filter uses a blacklist of 117 tokens that are classified as non-requirements. The list contains tokens that are often part of the requirement description but are not requirements in themselves. These include words like Wissen (eng. knowledge) and Erfahrung (eng. experience). The tokens were collected based on 50 job ads and on the token frequency.

The filter compares the stemmed blacklist tokens with the stemmed extracted skills. The stemming is used to filter out flexions of the blacklist words, too. If these match, the extracted token is discarded. For stemming the German Snowball Stemmer is used.

3.4.4 Semantic Matching

After the skill cleaning the set of extracted tokens contains only skills. The extracted skills can be syntactically different, but semantically similar or identical. For example:

```
IT-Sicherheit ← [IT-Security,
    Sicherheitsaspekte der IT]

GWT ← [Kenntnisse in GWT, Google
    Web Toolkit]
```

In order to get an overview of the most frequently required skills, exact matching is not relevant. Therefore, semantically similar requirements must be assigned to the same class. To this end, different text distance algorithms were compared. The evaluation of the algorithms is based on a test data set that contains 200 skills and relation to the correct target class. Two fast text models (Bojanowski et al., 2017; Grave et al., 2018)[3][4], one word2vec model[5] and four syntactic text distance algorithms were evaluated.

Table 6 shows the result of the comparison. With an accuracy of 44.00% the fasttext Model of Grave et al. (2018) achieves the best results, followed by the Levenshtein distance with 39.33%. The fasttext model recognizes the semantic relation between token like Diplom and Master. But especially abbreviations and inflections are mostly not recognized. Levenshtein provides good results when large character strings match. If only small parts of the string match, the Levenshtein

[3] Bojanowski et al. (2017): fasttext.cc/docs/en/pretrained-vectors

[4] Grave et al. (2018): fasttext.cc/docs/en/crawl-vectors

[5] devmount.github.io/GermanWordEmbeddings

Table 6: Comparing accuracy for text distance algorithms.

algorithm	accuracy
Fasttext (Grave et al. (2018))	0.4400
Levenshtein	0.3933
editex	0.3800
Hamming	0.2866
Fasttext (Bojanowski et al. (2017))	0.2133
Burrows–Wheeler transform Run-length encoding (BWT RLE)	0.1400
word2vec	0.1333

distance is also large. This leads to a wrong matching of extracted tokens, which contain a very short keyword like *C++* and other meaningless words like *knowledge*.

In order to take advantage of both, the fasttext model and the Levenshtein distance, both distances were combined and supplemented by preprocessing and stemming (see Algorithm 1). In preprocessing, the set of 511 skills S is extended by the respective stemmed skills. If the token t is contained in the set S, it is assumed that this is the best possible result. If not, a set $W = \{(w_0, p_0), ..., (w_n, p_n)\}$ of the 500 semantically most similar words w_i and the corresponding similarity p_i is calculated by the model m. It is assumed that stemmed(w_i) also has the similarity p_i to t (W'). From the set of all $w_i \in W'$ the intersection with S is formed ($W_\cap$). The similarity of each element of the intersection is weighted with the normalized Levenshtein distance and if t is a substring of w_i or w_i is a substring of t, p_i is weighted with $wc = 1.3$. The algorithm returns the word w_i with the greatest similarity p_i. This results in an accuracy of 75.33%.

3.5 Required Competencies

The algorithm was tested on 491 job ads. Figure 4 shows that professional experience, programming experience and a university degree are the most demanded competencies. The most important soft skills are the ability to work in a team, communication skills and a sense of responsibility. The most demanded programming languages are Java, C and Python. Linux, SAP and VMware lead the list with experience in handling concrete products.

4 CONCLUSIONS

Knowing the demands on employees is important for universities and students. Due to the lack of instruments to identify requirements for employees on the German labor market, a procedure was developed to automatically extract them from job advertisements. The procedure is subdivided into Data Acquisition, Language Detection, Section Classification and a multi-stage Skill Detection procedure. For data acquisition the online job portals monster.de, stepstone.de and stellenanzeigen.de were used as data sources. The language detection method according to Shuyo, based on n-grams, was used for speech recognition and reached an accuracy of 100% for the tested 300 job ads.

Algorithm 1: Semantic Matching.

```
Input: token to classify $t$, list of skills $S$, model $m$ with $m(x,n)$ returning
    sequence of n tuples of the most similar words to $x$ and the similarity.
Output: semantically most similar word $w_i$ to given token $t$
  $wc \leftarrow 1.3$
  $S \leftarrow S \cup \{stemmed(s) | s \in S\}$
  if $t \in S$ then
      return $t$
  else
      $W \leftarrow m(t, 500)$ // Result: $\{(w_0, p_0), ...(w_n, p_n)\}$
      $W' \leftarrow W \cup \{(stem(w_i), pi) | (w_i, p_i) \in W\}$
      $W_\cap \leftarrow \{(w_i, p_i) | (w_i, p_i) \in W' \wedge w_i \in S\}$
      $result_w, result_p = t, 0$
      for all $(w,p) \in W_\cap$ do
          if $w$ substring_of $t$ or $t$ substring_of $w$ then
              $p \leftarrow p * wc$
          end if
          $p \leftarrow p * (1 + norm(levens(w,t)))$
          if $p > result_p$ then
              $result_w, result_p = w, p$
          end if
      end for
      return $result_w$
  end if
```

To classify the sections, different methods of machine learning were compared. The best results with an accuracy of 0.9502 were achieved by the linear implementation of a Support Vector Machine using the one-vs-rest approach for multi-class problems. Taking into account the premise that per job advertisement only one HTML list element defines the employee requirements, an accuracy of 100% could be achieved.

The extraction of skills is based on natural language processing techniques. Using part of speech templates, tokens are extracted that correspond to the pattern of skills. False positives are filtered out via a blacklist. A synonym dictionary was created to correctly assign semantically identical or similar skills to a common class. This dictionary contains descriptions for skills in relation to semantically similar skills. Using a text distance algorithm based on fasttext word embedding and the Levenshtein distance the tokens are assigned to the skill classes. With this method, 75.33% of the known skills can be assigned correctly.

In the future, the semantic similarity recognition process could be improved. The inclusion of external data sources and training of word embeddings on

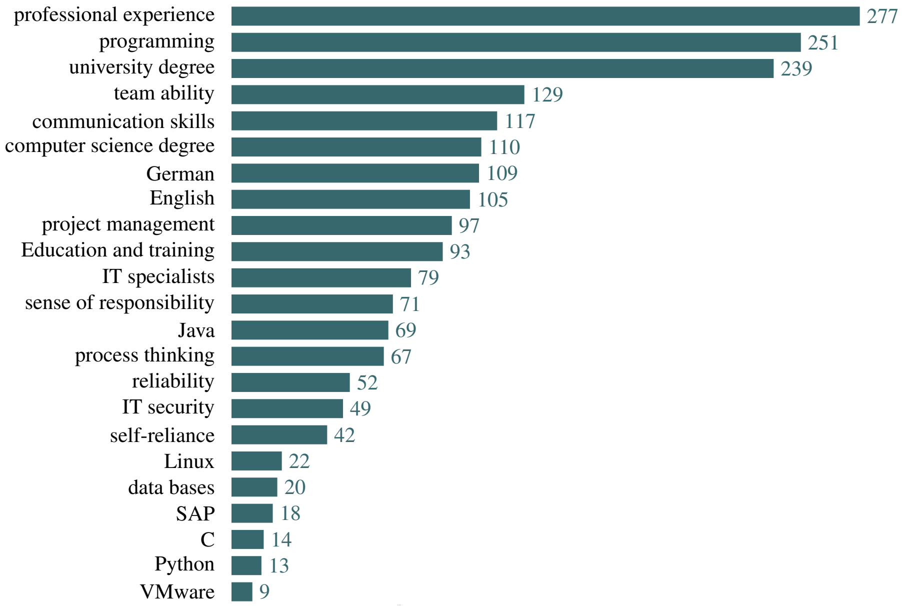

Figure 4: Most wanted competences based on 491 job ads (March 28, 2019).

job ads could improve the assignment. The procedure could also be extended to other occupational groups and, for example, provide up-to-date statistics online in real-time. In combination with other data in the job ads, maps showing the required skills in certain regions could be generated. In addition, the evaluation of the transferability of the approach to the English language and a comparison of the results would be interesting.

REFERENCES

Bitkom (2016). 51.000 offene stellen für it-spezialisten. Retrieved from bitkom.org/Presse/Presseinformation/51000-offene-Stellen-fuer-IT-Spezialisten.html.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Gallagher, K. P., Kaiser, K. M., Simon, J. C., Beath, C. M., and Goles, T. (2010). The requisite variety of skills for it professionals. *Commun. ACM*, 53(6):144–148.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

IMS (2018). Tiger corpus. Retrieved from ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html.

Kwon Lee, C. and Han, H. (2008). Analysis of skills requirement for entry-level programmer/analysts in fortune 500 corporations. *Journal of Information Systems Education*, 19.

Lee, S., Koh, S., Yen, D., and Tang, H.-L. (2002). Perception gaps between is academics and is practitioners: an exploratory study. *Information & Management*, 40(1):51–61.

Litecky, C., Aken, A., Ahmad, A., and Nelson, H. J. (2010). Mining for computing jobs. *IEEE Software*, 27(1):78–85.

Milton, T. (2000). Cross training the answer to e-commerce staff shortages. *Computer Weekly*.

Prabhakar, B., Litecky, C. R., and Arnett, K. (2005). It skills in a tough job market. *Communications of the ACM*, 48(10):91–94.

Scott, E., Alger, R., Pequeno, S., and Sessions, N. (2002). The skills gap as observed between is graduates and the systems development industry–a south african experience. *Informing Science*.

Shuyo, N. (2010). Language detection library for java. Retrieved from code.google.com/p/language-detection.

Sibarani, E. M., Scerri, S., Morales, C., Auer, S., and Collarana, D. (2017). Ontology-guided job market demand analysis. In Hoekstra, R., Faron-Zucker, C., Pellegrini, T., and de Boer, V., editors, *Proceedings of the 13th International Conference on Semantic Systems - Semantics2017*, pages 25–32, New York, New York, USA. ACM Press.

Weitzel, T., Eckhardt, A., Laumer, S., Maier, C., and Stetten, A. v. (2015). Recruiting trends 2015: Eine empirische untersuchung mit den top-1.000-unternehmen aus deutschland, sowie den top-300-unternehmen aus den branchen finanzdienstleistung, health care und it. Retrieved from nbn-resolving.de/urn:nbn:de:bvb:473-opus4-262833.

Wowczko, I. (2015). Skills and vacancy analysis with data mining techniques. *Informatics*, 2(4):31–49.

Yongbeom, K., Jeffrey, H., and Mel, S. (2006). An update on the is/it skills gap. *Journal of Information Systems Education*, 17(4):395–402.