Using Machine Learning for Labour Market Intelligence

Roberto Boselli[1,2], Mirko Cesarini[1,2], Fabio Mercorio[1,2], and Mario Mezzanzanica[1,2]

[1] Dept. of Statistics and Quantitative Methods, Univ. of Milano-Bicocca, Italy
[2] CRISP Research Centre, Univ. of Milano-Bicocca, Italy
(Applied Data Science Track: Data Analytics)

Abstract. The rapid growth of Web usage for advertising job positions provides a great opportunity for real-time labour market monitoring. This is the aim of Labour Market Intelligence (LMI), a field that is becoming increasingly relevant to EU Labour Market policies design and evaluation. The analysis of Web job vacancies, indeed, represents a competitive advantage to labour market stakeholders with respect to classical survey-based analyses, as it allows for reducing the time-to-market of the analysis by moving towards a fact-based decision making model. In this paper, we present our approach for automatically classifying million Web job vacancies on a standard taxonomy of occupations. We show how this problem has been expressed in terms of text classification via machine learning. Then, we provide details about the classification pipelines we evaluated and implemented, along with the outcomes of the validation activities. Finally, we discuss how machine learning contributed to the LMI needs of the European Organisation that supported the project.

Keywords: Machine Learning, Text Classification, Governmental Application

1 Introduction

In recent years, the European Labour demand conveyed through specialised Web portals and services has grown exponentially. This also contributed to introduce the term "Labour Market Intelligence" (LMI), that refers to the use and design of AI algorithms and frameworks to analyse Labour Market Data for supporting decision making. This is the case of *Web job vacancies*, that are job advertisements containing two main text fields: a *title* and a *full description*. The title shortly summarises the job position, while the full description field usually includes the position details and the relevant skills that the employee should hold.

There is a growing interest in designing and implementing real LMI applications to Web Labour Market data for supporting the policy design and evaluation activities through evidence-based decision-making. In 2010 the European Commission has published the communication "A new impetus for European

Cooperation in Vocational Education and Training (VET) to support the Europe 2020 strategy",[1] aimed at promoting education systems in general, and VET in particular. In 2016, the European Commission's highlighted the importance of Vocational and Educational activities, as they are "valued for fostering job-specific and transversal skills, facilitating the transition into employment and maintaining and updating the skills of the workforce according to sectorial, regional, and local needs".[2] In 2016, the EU and Eurostat launched the ESSnet Big Data project, involving 22 EU member states with the aim of "integrating big data in the regular production of official statistics, through pilots exploring the potential of selected big data sources and building concrete applications".

The rationale behind all these initiatives is that reasoning over Web job vacancies represents an added value for both *public and private* labour market operators to deeply understand the Labour Market dynamics, occupations, skills, and trends: (*i*) by reducing the time-to-market with respect to classical survey-based analyses (results of official Labour Market surveys actually require up to one year before being available); (*ii*) by overcoming the linguistic boundaries through the use of standard classification systems rather than proprietary ones; (*iii*) by representing the resulting knowledge over several dimensions (e.g., territory, sectors, contracts, etc) at different level of granularity and (*iv*) by evaluating and comparing international labour markets to support fact-based decision making.

Contribution. In this paper we present our approach for classifying Web job vacancies, we designed and realised within a research call-for-tender[3] for the Cedefop EU organisation[4]. Specifically, the goal of this project was twofold: first, the evaluation of the effectiveness of using Web Job vacancies for LMI activities through a feasibility study, second, the realisation of a working prototype that collects and analyses Web job vacancies over 5 Countries (United Kingdom, Ireland, Czech Republic, Italy, and Germany) for obtaining near-real time labour market information. Here we focus on the classification task showing the performances achieved by three classification pipelines we evaluated for realising the system.

We begin by discussing related work in Sec. 2. In Sec. 3 we discuss how the problem of classifying Web job vacancies has been solved through machine learning, providing details on the feature extraction techniques used. Sec. 4 provides the experimental results about the evaluation of three distinct pipelines employed. Sec. 5 concludes the paper and describes the ongoing research.

---

[1] Publicly available at https://goo.gl/Goluxo

[2] The Commission Communication "A New Skills Agenda for Europe" COM(2016) 381/2, available at https://goo.gl/Shw7bI

[3] "Real-time Labour Market information on skill requirements: feasibility study and working prototype". Cedefop Reference number AO/RPA/VKVET-NSOFRO/Real-time LMI/010/14. Contract notice 2014/S 141-252026 of 15/07/2014 https://goo.gl/qNjmrn

[4] Cedefop European agency supports the development of European Vocational Education and Training (VET) policies and contributes to their implementation - http://www.cedefop.europa.eu/

2 Related Work

Labour Market Intelligence is an *emerging* cross-disciplinary field of studies that is gaining research interests in both *industrial* and *academic* communities.

Industries. Information extraction from unstructured texts in the labour market domain mainly focused on the e-recruitment process (see, e.g.,[19]) attempting to support or automate the *resume* management by matching candidate profiles with job descriptions using machine learning approaches [30,32,11]. Concerning companies, their need to automatize Human Resource (HR) department activities is strong; as a consequence, a growing amount of commercial skill-matching products have been developed in the last years, for instance, Burning-Glass, Workday, Pluralsight, EmployInsight, and TextKernel. To date, the only commercial solution that uses international standard taxonomies is Janzz: a Web based platform to match labour demand and supply in both public and private sectors. It also provides APIs access to its knowledge base, but it is not aimed at classifying job vacancies. Worth of mentioning is Google Job Search API, a pay-as-you-go service announced in 2016 for classyfying job vacancies through the Google Machine Learning service over O*NET, that is the US standard occupation taxonomy. Though this commercial service is still a closed alpha, it is quite promising and also sheds the light on the needs for reasoning with Web job vacancies using a common taxonomy as baseline.

Literature. Since the early 1990s, *text classification* (TC) has been an active research topic. It has been defined as "the activity of labelling natural language texts with thematic categories from a predefined set" [29]. Most popular techniques are based on the *machine learning* paradigm, according to which an automatic text classifier is created by using an inductive process able to learn, from a set of pre-classified documents, the characteristics of the categories of interest.

In the recent literature, text classification has proven to give good results in categorizing many real-life Web-based data such as, for instance, news and social media [15,33], and sentiment analysis [20,25]. To the best of our knowledge, text classifiers have not been applied yet to the classification of Web job vacancies published on several Web sites for analysing the Web job market of a geographical area, and this system is the first example in this direction.

All these approaches are quite relevant and effective, and they also make evidence of the importance of the Web for labour market information. Nonetheless, they differ from our approach in two aspects. First, we aim to classify *job vacancies* according to a target classification system for building a (language independent) knowledge base for analyses purposes, rather than matching resumes on job vacancies. Furthermore, resumes are usually accurately written by candidates whilst Web advertisements are written in a less accurate way, and this quality issue might have unpredictable effects on the information derived from them (see, e.g. [22,21,6,9] for practical applications). Second, the system aims at producing analyses based on standard taxonomies to support the fact-based decision making activities of several stakeholders.

3 Text Classification in LMI

The Need for a Standard Occupations Taxonomy. The use of proprietary and language-dependent taxonomies can prevent the effective monitoring and evaluation of Labour Market dynamics across national borders. For these reasons, a great effort has been made by International organisations for designing *standard* classifications systems, that would act as a lingua-franca for the Labour Market to overcome the linguistic boundaries as well. One of the most important classification system designed for this purposes is ISCO: The *International Standard Classification of Occupations* has been developed by the International Labour Organization as a four-level classification that represents a standardised way for organising the labour market occupations. In 2014, ISCO has been extended through ESCO: the multilingual classification system of European Skills, Competences, Qualifications and Occupations, that is emerging as the European standard for supporting the whole labour market intelligence over 24 EU languages. Basically, the ESCO data model includes the ISCO hierarchical structure as a whole and extends it through a taxonomy of skills, competences and qualifications.

3.1 The Classification Task.

Text categorisation aims at assigning a Boolean value to each pair $(d_j, c_i) \in D \times C$ where D is a set of documents and C a set of predefined categories. A *true* value assigned to (d_j, c_i) indicates document d_j to be set under the category c_i, while a false value indicates d_j cannot be assigned under c_i. In our scenario, we consider a set of job vacancies $\mathcal{J}$ as a collection of documents each of which has to be assigned to one (and only one) ISCO occupation code. We can model this problem as a text classification problem, relying on the definition of [29].

Formally speaking, let $\mathcal{J} = \{J_1, \ldots, J_n\}$ be a set of job vacancies, the classification of $\mathcal{J}$ under the ESCO classification system consists of |O| independent problems of classifying each job vacancy $J \in \mathcal{J}$ under a given ESCO occupation code o_i for $i = 1, \ldots, |O|$. Then, a *classifier* is a function $\psi : \mathcal{J} \times O \to \{0, 1\}$ that approximates an unknown target function $\dot{\psi} : \mathcal{J} \times O \to \{0, 1\}$. Clearly, as we deal with a single-label classifier, $\forall j \in \mathcal{J}$ the following constraint must hold: $\sum_{o \in O} \psi(j, o) = 1$.

In this paper, job vacancies are classified according to the 4^{th} level of the ISCO taxonomy (and the corresponding multilingual concepts of the ESCO ontology) as further detailed in the next sections. The choice of the ISCO 4^{th} level (also referred as ISCO 4 digits classification) is a trade-off between the granularity of occupations (the more digits the better) and the effort to develop an automatic classifier (the fewer digits the better). The job vacancy classification is translated into a supervised machine learning text classification problem, namely a multiclass single label classification problem i.e., a job offer is classified to one and only one 4 digits ISCO code over a set of 436 available ones.

Within this project we decided to use *titles* for occupation classification. Indeed, in our very preliminary studies [2] we experimentally observed that titles

are often concise and highly focused on describing the proposed occupations while other topics are hardly dealt, making titles suitable for the classification task.

3.2 Feature Extraction

Two feature extraction methods have been evaluated for classifying job occupation, namely: Bag of Word Approach, and Word2Vec, that we describe in the following.

Bag of Word Feature Extraction. Titles were pre-processed according to the following steps: (*i*) html tag removal, (*ii*) html entities and symbol replacement, (*iii*) tokenization, (*iv*) lower case reduction, (*v*) stop words removal (using the stop-words list provided by the NLTK framework [5]), (*vi*) stemming (using the Snowball stemmer), (*vii*) n-grams frequency computation (actually, unigram and bigram frequencies were computed, n-grams which appear less than 4 times or that appear in more than 30% of the documents are discarded, since they are not significant for classification). Each title is pre-processed according to the previous steps and is transformed into a set of n-gram frequencies.

Word2Vec Feature Extraction. Each word in a title was replaced by a corresponding vector of an n-dimensional space. We used a vector representation of words belonging to the family of neural language models [3] and specifically we used the Word2Vec [23,24] representation.

In neural language models, every word is mapped to a unique vector, given a word w and its context k (n words nearby w), the concatenation or sum of the vectors of the context words is then used as features for prediction of the word w [24]. This problem can be viewed as a machine learning problem where n context words are fed into a neural network that should be trained to predict the corresponding word, according to the Continuous Bag of Words (CBOW) model proposed in [23].

The word vector representations are the coefficient of the internal layers of the neural network, for more details the interested reader can refer to [24]. The word vectors are also called word embeddings.

After the training ends, words with similar meaning are mapped to a similar position in the vector space [23]. For example, “powerful” and “strong” are close to each other, whereas “powerful” and “Paris” are more distant. The word vector differences also carry meaning.

We used the GENSIM [27] implementation of Word2Vec to identify the vector representations of the words. Since Word2Vec requires huge text corpora for producing meaningful vectors, we used all the downloaded job vacancies to train the Word2Vec (the *unlabelled dataset*, about 6 Million of job vacancies, as outlined in Sec. 4.1). The 6 Million job vacancy texts underwent the steps from (*i*) to (*vi*) of the processing pipeline described in Subsec. 3.2 before being used for training the Word2Vec model. Actually, the Wod2Vec model was trained using vectors of size equals to 300 using the CBOW training algorithm.

The Word2Vec embeddings were used to process the titles of the labelled dataset introduced in Sec. 4.1 as follows: steps from (*i*) to (*vi*) of the processing pipeline described in Subsec. 3.2 were executed on titles. The first 15 tokens of titles where considered (i.e., tokens exceeding the 15^{th} were dropped, as the affected titles account for less than 0.2% of total vacancies). Each word in the title was replaced by the corresponding (word) vector, e.g., given a set of n titles each one composed by 15 words, the output of the substitution can be viewed as a 3-dimensional array (e.g., a 3-dimensional matrix or a 3-dimensional tensor) of the shape: [n_documents, 15, word_vector_dimension].

4 Experimental Results

This section introduces the evaluation performed on the several classification pipelines and the dataset used.

4.1 Datasets

Two datasets have been considered in the experiments outlined in this section:

Labelled A set of 35,936 job vacancies manually labelled using 4 digits ISCO code. Not all the 4 digits ISCO occupations are present in the dataset, only 271 out of 436 ISCO codes were actually found. It is worth to mention that ISCO tries to categorize all possible occupations, but some are hardly found on the Web (e.g., 9624 *Water and firewood collectors*[5]). The interested reader can refer to [13] for further information.

Unlabelled A set of 6,005,916 unlabelled vacancies. The vacancies have been collected for one year scraping 7 web sites focusing on the UK and Irish Job Market. For each vacancy both a title and a full description is available.

The *labelled* dataset was used to train a classifier to be used later to identify the ISCO occupations on the *unlabelled* vacancy dataset. The latter was used to compute the Word2Vec word embeddings.

In the following sections, the classification pipelines we have evaluated are introduced. For evaluation purposes, the labelled dataset was randomly split into train and test (sub)sets containing respectively 75% and 25% of the vacancies. The vacancies of each ISCO code were distributed in the two subsets using the same proportions.

4.2 Classification pipelines

This subsection will introduce the several classification pipelines which have been evaluated for classifying job vacancies. Each pipeline has parameters whose optimal values have been found performing a Grid Search as detailed in Sec. 4.3.

[5] Tasks include cutting and collecting wood from forests for sale in market or for own consumption ... drawing water from wells, rivers or ponds, etc. for domestic use.

BoW - SVM. The BoW feature extraction pipeline (described in Sec. 3.2) was used on the (labelled) training dataset and the results were used to feed two classifiers, namely Linear SVM and Gaussian SVM, the latter also known as radial basis function (RBF) SVM kernel [8]. They will be called LinearSVM and RBF SVM hereafter.

According to [14], SVM is well suited to the particular properties of texts, namely high dimensional feature spaces, few irrelevant features (dense concept vector), and sparse instance vectors. The parameters evaluated during the grid search are $C \in \{0.01, 0.1, 1, 10, 100\}$ for LinearSVM classifier, while $C \in \{0.01, 0.1, 1\} \times$ Gamma $\in \{0.1, 1, 10\}$ for RBF SVM.

BoW - Neural Network. The BoW feature extraction pipeline (described in Sec. 3.2) was also used to feed the fully connected neural networks described below. Each neural network has an input of size 5,820 (the number of features produced by the feature extraction pipeline) and an output of size 271 (the number of ISCO codes in the training set). Each layer (if not otherwise specified) use the Linear Rectifier as non linearity, excluding the last layer which uses Softmax. In the networks described below each fully connected layer is preceded by a batch normalization layer [12], whose purpose is to accelerate the network training (and it doesn't have an effect on the classification performances).

- (FCNN1) is a 4 layer neural network having 2 hidden layers: a batch normalization and a fully connected layer of size $3,000$.
- (FCNN2) is a 5 layer neural network, having 4 hidden layers: two fully connected layers respectively of $3,900$ and $2,000$ neuron size, each one preceded by a batch normalization layer.

Word2Vec - Convolutional Neural Networks. Convolutional Neural Networks (CNNs) are a type of Neural Network where the first layers act as filters to identify patterns in the input data set and consequently to work on a more abstract representation of the input. CNNs were originally employed in Computer Vision, the interested reader can refer to [18,17,28] for more details. CNNs have been employed to solve text classification tasks [16,7].

In this paper, we evaluated the convolutional neural network described in Tab. 1 over the results of the Word2Vec pipeline described in Sec. 3.2.

The first two convolutional layers perform a convolution over the Word2Vec features producing as output respectively the results of 200 and 100 filters. At the end of the latter convolutional layer, each title can be viewed as a matrix of 15×100 (respectively, the number of words and the quantity of filter values computed on each word embedding). The FeaturePoolLayer averages the 15 values for each filter, at the end of this layer each title can be viewed as a vector of 100 values. Then, two fully connected layers follow of respectively $2,000$ and 500 neurons (each fully connected layer is preceded by a Batch Normalisation Layer). The last layer has as many neurons as the number of ISCO codes available in the training set and employs softmax as non linearity.

The network was trained using Gradient Descent, the network accounts for about 10^6 weights to be updated during the training. It wasn't necessary splitting

Table 1: The Convolutional Neural Networks Structure. Each layer non linearity is specified in the note (if any). A BatchNormLayer performs a batch normalization

Network Layer	Layer Type	Note
1	Input Layer	input shape: [n_documents, 15, word_vector_dimension]
2	Conv1DLayer	num_filters=200, filter_size=1, stride=1, pad=0, nonlinearity=linear rectifier
3	Conv1DLayer	num_filters=100, filter_size=1, stride=1, pad=0, nonlinearity=linear rectifier
4	FeaturePoolLayer	for each filter it computes the mean across the 15 word values
5	Fully Connected Layer	num_units=2000, nonlinearity=linear rectifier
6	BatchNormLayer	
7	Fully Connected Layer	num_units=500, nonlinearity=linear rectifier
8	BatchNormLayer	
9	Fully Connected Layer	The final layer, nonlinearity=softmax

the documents into batches during the training since the GPU memory was enough to handle all of them. The initial learning rate was 0.1 and Nesterov Momentum was employed. Early stopping was used to guess when to stop the training.

4.3 Experimental Settings

The classification pipelines previously introduced have been evaluated using the train and test datasets on which the labelled dataset was split into. The unlabelled dataset was used to train the Word2Vec model, which was then employed in the feature extraction process over the labelled dataset. The extracted features were used to perform a supervised machine learning process. Each classification pipeline has parameters requiring tuning, therefore a grid search was performed on the train set using a k-fold cross validation (k=5) to identify the combination of parameters maximizing the F1-score (actually it was used the weighted F1-score). For each classification pipeline, the best combination of parameters was evaluated against the test set. The results are outlined in the remaining of this section.

The classifiers were built using the Scikit-learn [26], Theano [31], and Lasagne [10] frameworks running on an Intel Xeon machine with 32GB Ram and an NVidia CUDA 4GB GPU. Considering the BoW Feature Extraction, the Linear SVM classifier parameters and performances are shown in Tab. 2a, the SVM RBF performances are shown in Tab. 2c, and the Fully Connected Neural Network classifier performances are shown in Tab. 2b.

Results Summary Tab. 3 summarises the best parameters for each classification pipeline and outlines the performance computed on the test set.

The BoW SVM Linear has the best performances, therefore it has been chosen for implementing the occupation classification pipeline for the English language in the prototype. As stated in the literature, text classification can be efficiently solved using linear classifiers [14] like Linear SVM, and the additional complexity of non-linear classification does not tend to pay for itself [1], except

Table 2: Classification pipelines parameters and performances. NGram Range focuses on BoW feature extraction: (1,1) is for *Only Unigrams*, (1,2) is for *Both Unigrams and Bigrams*. The F1 score, precision, and recall are the weighted average of the corresponding scores computed on each ISCO code class. In Tab. (C) only a subsets of the results is shown. Grid search was computed using a 5-fold cross validation on the training set

(a) BoW - LinearSVM

C	NGram Range	F1-S	Prec	Rec
0.01	(1, 1)	0.786	0.798	0.797
100	(1, 1)	0.797	0.806	0.793
100	(1, 2)	0.816	0.826	0.813
10	(1, 1)	0.825	0.831	0.825
0.01	(1, 2)	0.834	0.842	0.838
10	(1, 2)	0.835	0.842	0.833
1	(1, 1)	0.845	0.849	0.846
0.1	(1, 1)	0.846	0.851	0.849
1	(1, 2)	0.854	0.858	0.854
0.1	**(1, 2)**	**0.858**	**0.862**	**0.859**

(b) BoW - Neural Net.

NGram Range	Net	F1-S	Prec	Rec
FCNN1	(1, 1)	0.778	0.786	0.783
FCNN2	(1, 1)	0.784	0.790	0.783
FCNN2	(1, 2)	0.801	0.809	0.799
FCNN1	**(1, 2)**	**0.816**	**0.822**	**0.818**

(c) BoW - RBF SVM

F1-S	Prec	Rec	C	γ	Ngram Range
0.016	0.049	0.025	0.01	10	(1, 1)
0.018	0.061	0.023	0.01	10	(1, 2)
0.020	0.033	0.029	0.01	0.1	(1, 1)
0.027	0.049	0.036	0.01	1	(1, 2)
0.028	0.038	0.039	0.01	0.1	(1, 2)
...	...	...	...	...	...
0.718	0.849	0.644	100	1	(1, 1)
0.827	0.840	0.822	100	0.1	(1, 1)
0.831	0.845	0.825	100	0.1	(1, 2)
0.836	0.852	0.832	1	0.1	(1, 1)
0.840	0.851	0.836	10	0.1	(1, 1)
0.841	0.854	0.836	10	0.1	(1, 2)
0.842	**0.861**	**0.835**	**1**	**0.1**	**(1, 2)**

Classification Pipeline	Notes	F1 Score	Precision	Recall
BoW SVM Linear	C=0.1, NGram_Range=(1,2)	**0.857**	0.870	**0.865**
BoW SVM RBF	C=1, Gamma=0.1, NGram_Range=(1,2)	0.849	**0.878**	0.856
BoW Neural Network	Net=FCNN1, NGram_Range=(1,2)	0.820	0.835	0.830
W2V CNN	Net=CNN	0.787	0.802	0.797

Table 3: Classification pipelines performances computed on the test dataset

for some special data sets. Considering the Word2Vec Convolutional Neural Network, the performances are shown in Tab. 3, the authors would have expected better results, and the matter calls for further experiments.

4.4 Results Validation by EU Organisation

The project provided two main outcomes to the Cedefop EU Agency that supported it. On one side, a *feasibility study*, that has not been addressed in this paper, reporting some best practices identified by labour market experts involved within the project and belonging to the ENRLMM.[6] As a major result, the project provided a *working prototype* that has been deployed at Cedefop in June 2016 and it is currently running on the Cedefop Datacenter. In Fig. 1 we report a snapshot from a demo dashboard that provides an overview of the occupation trends over the period June-September 2015 collecting up to 700K unique Web job vacancies. To date, the system collected 7+ million job vacancies over the 5 EU countries, and it is among a selection of research projects of Italian universities framed within a Big Data context [4].

Below we report an example of how the classified job vacancies could be used to support LMI monitoring, focusing only on 4-months scraping data. One might closely looks at the differences between countries in terms of labour market demands. Comparing UK against Italy, we can see that "Sales, Marketing and development managers" are the most requested occupations at ESCO (level 2) in the UK over this period whilst, rolling up on ESCO level 1 we can observe

[6] The Network on Regional Labour Market Monitoring, http://www.regionallabourmarketmonitoring.net/

that "Professionals" are mainly asked in the "Information and Communication" sector, followed by "Administrative and support service activities" according to the NACE taxonomy. Furthermore, the type of contract is usually specified offering permanent contracts. Differently, the Italian labour market, that has a job vacancy posting rate ten times lower than UK over the same period, is looking at Business Service Agents requested in the "Manufacturing" field offering often temporary contracts.

Interactive Demos. Due to the space restrictions, the dashboard in Fig.1 and some other demo dashboards have been made available online: *Industry and Occupation Dashboard* at https://goo.gl/bdqMkz, *Time-Series Dashboard* at https://goo.gl/wwqjhz, and *Occupations Dashboard* at https://goo.gl/M1E6x9

Project Results Validation. Finally, the project results have been discussed and endorsed in a workshop which took place in Thessaloniki, in December 2015.[7] The methodology and the results obtained have been validated as effective by leading experts on LMI and key stakeholders. In 2017, we have been granted by Cedefop the extension of the prototype to all the 28 EU Countries.[8]

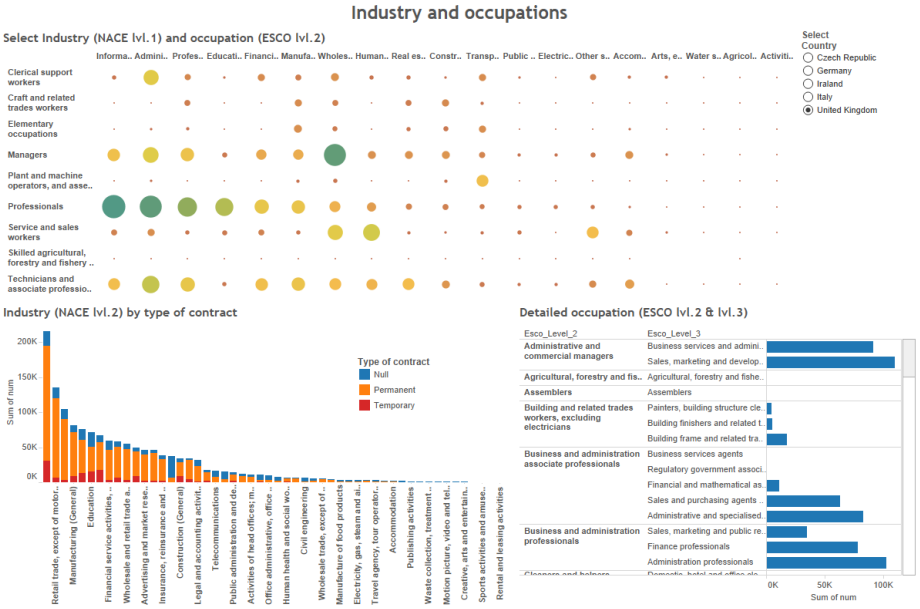

Fig. 1: A snapshot from the System Dashboard deployed. Interactive Demo available at https://goo.gl/bdqMkz

[7] The Workshop agenda and participants list is available at https://goo.gl/71Oc7A

[8] "Real-time Labour Market information on Skill Requirements: Setting up the EU system for online vacancy analysis AO/DSL/VKVET-GRUSSO/Real-time LMI 2/009/16. Contract notice - 2016/S 134-240996 of 14/07/2016 https://goo.gl/5FZS3E

5 Conclusions and Expected Outcomes

In this paper we have described an innovative real-world data system we developed within a European research call-for-tender, granted by a EU organisation, aimed at classifying Web job vacancies through machine learning algorithms. We designed and evaluated several classification pipelines for assigning ISCO occupation codes to job vacancies, focusing on the English language. The classification performances guided the implementation of similar pipelines for different languages. The main outcome of this project is a working prototype actually running on the Cedefop European Agency datacenter, collecting and classifying Web job vacancies from 5 EU Countries. The developed system provides an important contribution to the whole LMI community and it is among the first research projects that employed machine learning algorithms for obtaining near real-time information on Web job vacancies. The results have been validated by EU labour market experts and put the basis of a further call to extend the system to all the EU Countries, which we are currently working on.

References

1. Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Mining text data, pp. 163–222. Springer (2012)
2. Amato, F., Boselli, R., Cesarini, M., Mercorio, F., Mezzanzanica, M., Moscato, V., Persia, F., Picariello, A.: Challenge: Processing web texts for classifying job offers. In: Semantic Computing (ICSC), 2015 IEEE International Conference on. pp. 460–463 (2015)
3. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research 3(Feb), 1137–1155 (2003)
4. Bergamaschi, S., Carlini, E., Ceci, M., Furletti, B., Giannotti, F., Malerba, D., Mezzanzanica, M., Monreale, A., Pasi, G., Pedreschi, D., et al.: Big data research in italy: A perspective. Engineering 2(2), 163–170 (2016)
5. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
6. Boselli, R., Mezzanzanica, M., Cesarini, M., Mercorio, F.: Planning meets data cleansing. In: The 24th International Conference on Automated Planning and Scheduling (ICAPS 2014). pp. 439–443. AAAI (2014)
7. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: The 25th International Conference on Machine Learning. pp. 160–167. ICML, ACM (2008)
8. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning 20(3), 273–297 (1995)
9. Dasu, T.: Data glitches: Monsters in your data. In: Handbook of Data Quality, pp. 163–178. Springer (2013)
10. Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S.K., Nouri, D., et al.: Lasagne: First release. (Aug 2015), http://dx.doi.org/10.5281/zenodo.27878
11. Hong, W., Zheng, S., Wang, H.: Dynamic user profile-based job recommender system. In: Computer Science & Education (ICCSE). IEEE (2013)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
13. International standard classification of occupations (2012), visited on 2016-11-11
14. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) Machine Learning: ECML-98, Lecture Notes in Computer Science, vol. 1398, pp. 137–142. Springer (1998)

15. Khan, F.H., Bashir, S., Qamar, U.: Tom: Twitter opinion mining framework using hybrid classification scheme. Decision Support Systems 57, 245 – 257 (2014)
16. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
18. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: The 26th Annual International Conference on Machine Learning. pp. 609–616. ICML '09, ACM (2009)
19. Lee, I.: Modeling the benefit of e-recruiting process integration. Decision Support Systems 51(1), 230–239 (2011)
20. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2009)
21. Mezzanzanica, M., Boselli, R., Cesarini, M., Mercorio, F.: Data quality sensitivity analysis on aggregate indicators. In: Helfert, M., Francalanci, C., Filipe, J. (eds.) Proceedings of the International Conference on Data Technologies and Applications, Data 2012, pp. 97–108. INSTICC (2012)
22. Mezzanzanica, M., Boselli, R., Cesarini, M., Mercorio, F.: A model-based evaluation of data quality activities in KDD. Information Processing & Management 51(2), 144–166 (2015)
23. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
25. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)
26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
27. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
28. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks 61, 85 – 117 (2015)
29. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1), 1–47 (2002)
30. Singh, A., Rose, C., Visweswariah, K., Chenthamarakshan, V., Kambhatla, N.: Prospect: a system for screening candidates for recruitment. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 659–668. ACM (2010)
31. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688 (May 2016), http://arxiv.org/abs/1605.02688
32. Yi, X., Allan, J., Croft, W.B.: Matching resumes and jobs based on relevance models. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 809–810. ACM (2007)
33. Zubiaga, A., Spina, D., Martínez-Unanue, R., Fresno, V.: Real-time classification of twitter trends. JASIST 66(3), 462–473 (2015)