This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Information Extraction from Free-Form CV Documents in Multiple Languages

DAVOR VUKADIN, ADRIAN SATJA KURDIJA (Member, IEEE), GORAN DELAČ (Member, IEEE), and MARIN ŠILIĆ, (Member, IEEE)
Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

Corresponding author: Marin Šilić (e-mail: marin.silic@fer.hr).

The authors would like to thank the Ework Group for providing the dataset. The authors acknowledge the support of European Regional Development Fund through the *System for Detection of Malicious Transactions in Electronic Payment Operations Based on Machine Learning* research project (IRI-II KK.01.2.1.02.0192) and the *VODIME - The Waters of Imotski Region* research project (KK.05.1.1.02.0024). The authors acknowledge the support of Croatian Science Foundation through the *Reliable Composite Applications Based on Web Services* (HRZZ-IP-01-2018-6423) research project. The Titan X Pascal used for this research was donated by NVIDIA Corporation.

ABSTRACT This paper proposes two natural language processing models for extracting useful information from multilingual, unstructured (free form) CV documents. The model identifies the relevant document sections (personal information, education, employment, etc.) and the corresponding specific information at the lower hierarchy level (names, addresses, roles, skill competences, etc.). Our approach employs the transformer architecture and its multilingual implementation of the encoder part in the form of the BERT language model. The models are trained and tested on a large, manually annotated CV dataset, achieving high scores on standard accuracy measures. The proposed models exhibit important properties of end-to-end training and interpretability, which was investigated by visualizing the model attention and its vector representations.

INDEX TERMS Information retrieval, Natural language processing, Text analysis, Recurrent neural networks, CV parsing

I. INTRODUCTION

Automatic extraction of useful information from CVs given in free form is a difficult task in the area of natural language processing (NLP). A system which could convert a free-form CV into a given highly organized structure can be a very valuable tool to recruiters and various job market websites. Useful information in this case includes personal information such as first and last name, residential addresses and spoken language, as well as information about past employments, education and skills or competences of the person. As any of this information can be presented in many different formats, it is not possible to create a simple parser (e.g. using regular expressions) that would accurately extract all the important information. The problem becomes even more complicated when multiple languages have to be supported, which is another obstacle for a potential rule-based parser.

Therefore, in our work, machine learning techniques are used in the context of NLP in order to achieve a high degree of accuracy in extracting the desired information in arbitrary format in five languages. Our work addresses the CV parsing problem by constructing an NLP system with several interconnected machine learning models, using state-of-the-art NLP models as the basis. Namely, common approaches in NLP involve the use of recurrent neural networks that receive the elements of the input sequence and perform classification for them. A new deep model architecture recently designed for sequential input data, called *transformer* ([1]), allows parallel processing of the input sequence. It also allows a certain degree of model interpretability due to the attention layers used as the main element of feature extraction. Our approach uses the transformer architecture and its multilingual implementation of the encoder part in the form of the BERT language model. The model extracts and classifies the relevant *sections* of a document (personal information, education, past employment, skills) and, at a lower hierarchy level, extract and classifies the corresponding specific information such as names, dates, organizations, positions, university degrees, individual skills and their (self-assessed) competence degrees.

The models were evaluated using standard metrics: pre-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

cision, recall and F1 scores on a dataset consisting of 1686 annotated CVs in five languages: English, Swedish, Norwegian, Finnish and Polish. The languages come from various language families and sub-families, which highlights the effectiveness of our proposed method. Finnish is a member of the Uralic language family, while the rest of the languages belong to the Indo-European family. Swedish and Norwegian are members of the Indo-European North Germanic language sub-family, English, of the West Germanic sub-family and finally, Polish belongs to the Slavic sub-family. The original CVs were obtained from a Northern European independent consultant provider - Ework Group[1].

The main contributions of this work are listed below.

- Categories and labels for efficient classification of the CV information are defined.
- A multilingual model for extraction of section and item level information from a CV document is described.
- A model for extraction of self-assessment competence information for individual skills is described.
- The proposed models are thoroughly tested and analyzed, using data in multiple languages: English, Swedish, Norwegian, Polish, and Finnish.

The paper is organized as follows. Sect. II comments on related work and models which we use as part of the proposed system. Sect. III describes the proposed models along with the labels proposed for identification of important CV information. Sect. IV describes detailed evaluation results including accuracy measures. Sect. V gives visualizations of attention and vector representation for both proposed models. Conclusions are given in Sect. VI.

II. RELATED WORK

This section gives a brief overview of the relevant previous research done in the areas of natural language processing and CV parsing. In Sect. II-A we consider the state-of-the-art NLP models that we have applied for the purposes of CV parsing. In Sect. II-B we discuss alternative CV parsing approaches proposed by other researchers.

A. RELEVANT NLP MODELS

Recurrent models that primarily use *long short-term memory* ([2]) and *gated recurrent* cells ([3]) are considered state-of-the-art in many areas of natural language processing – sequence and language modeling, and machine translation. Recurrent models typically divide the calculation to each segment of the input string. On each input position, the hidden state h_t is calculated as a function of the previous state h_{t-1} and the current input to the model at position t. This way of array processing disables parallelization, which becomes a problem with long input sequences, where limited memory reduces the ability to use mini-batches.

Attention mechanisms have become an integral part of recurrent models, enabling the modelling of mutual dependence of different segments of the input sequence regardless of their mutual distance. The authors in [1] proposed a new architecture called transformer, which instead of the recurrent mechanism exclusively uses the attention mechanism to model global dependencies between model input and output. This allowed the parallelization of the models which deal with sequences and new state-of-the-art performance has been set in a number of NLP areas. The attention mechanism can be described as mapping of a query and a collection of key-value pairs to the output, where query, key, value, and output are vectors. The output is calculated as the weighted mean of all values, where the weight of each value is determined by compatibility of the query to the respective key. Saving all inputs in one matrix allows parallel attention calculation for all input elements.

Bidirectional Encoder Representations from Transformers (BERT) is a model presented in [4]. Unlike earlier models designed for language modeling, BERT is designed to learn deep bidirectional representations on unlabeled text by joint conditioning using left and right contexts in all layers of the model. This allows fine-tuning of BERT parameters on a large number of other problems. WordPiece ([5]) vector representations with a vocabulary of 30 000 tokens are used as input to BERT. The first token of each sequence is a special classification token (**[CLS]**) and the final hidden state corresponding to this token is used as an aggregate representation of the entire input sequence for classification problems. The BERT architecture is actually the multilayer encoder of the transformer. In this work we have used a multilingual model that is case sensitive, pre-trained with a corpus of 104 different languages.

B. CV PARSING

Standard methods of extracting useful information from sources such as CV documents, where the structure varies from one document to another, are usually costly in terms of time and resources for manual rule design or a careful design of features which would enable development of a machine learning algorithm. The problem with these approaches is their non-robustness to unprecedented examples, for which it is then necessary to adjust the conceived rules or manually created features. More weaknesses of such approaches arise when there is a need to support multiple CV languages, where each change in the logic of the extractor needs to be mapped to the required number of languages.

Researchers have investigated the resume parsing problem with different assumptions, goals and results. The work in [6] uses hierarchical labels similar to our work (with section and item levels), but separate models for each level are used, so the training is not end-to-end as in our approach. Instead, first the section level is tagged using a hidden Markov model (HMM) ([7]), then the *Education* section is sent to another HMM, and the *Personal* section to the SVM model, and these models then tag each element using term frequency - inverse term frequency (TFIDF) ([8]) input features.

The work in [9] uses a combination of Convolutional Neural Network (CNN) ([10]), Conditional Random Field (CRF)

[1]https://www.eworkgroup.com/

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

([11]) and Bidirectional Long-Short-Term-Memory - CNN (Bi-LSTM-CNN) ([2]) models for sequence tagging with hierarchical labels. Global Vectors for Word Representation (GloVe) ([12]) word embeddings are used, and a CNN model first divides a CV into *Personal*, *Occupation* and *Educational* sections. Then the output of each section is passed through three different CRF + Bi-LSTM-CNN models. Again, the advantage of our work is the use of end-to-end training and our model has the ability to use cross-information between section and item levels.

To name some less relevant approaches, parsing by [13] is viewed as a simple sequence tagging problem (without hierarchical labels), for which the word2vec embeddings ([14]) and the CRF model were used. The authors have also tried using hand-crafted features, but with worse performance results than when using word2vec. In [15] a manual parser was created for important features, but not all feature examples are given and the parser was not described. In [16], the keywords from the database are hard-matched and the score with respect to the job description is calculated. Similarly, in [17] a candidate-job score is created using hard string matching of skills and institutions from the database.

An interesting approach in [18] uses heuristics for dividing a CV document into chunks based on the visual features from the document in PDF format (font type and size, spaces, etc.). The chunks were classified using a Support Vector Machine (SVM) ([19]) model; then the *Education* and *Personal* chunks were given to the CRF to extract the detailed information. Similar to [6] and our work, the hierarchical labels are used, but model is not trained end-to-end and manually designed features were used for both SVM and CRF.

It is important to note that none of the above mentioned papers dealt with multilingual data. Also, their models were not transparent in the sense of the ability to understand why some outputs were chosen instead of the others, which is an important property of our models (see Sect. V).

III. CV INFORMATION EXTRACTION

This section describes the proposed models for extracting information from free-form CVs. First, in Sect. III-A we propose a classification of parts of the CV text which is suitable for automatic extraction of useful information. Sect. III-E describes in detail the architecture of the models, while the training parameters are specified in Sect. III-F.

A. TYPES OF INFORMATION

We have defined two levels of information: *item level* and *section level*. Item level contains "hard" information such as names (of a person or organizations such as previous employers), locations, dates etc. The purpose of the section level is to contextualize the item level information. For example, a university name (detected in the item level) can belong to an education section (if the person studied there) or to an employment section (if the person worked there, e.g. as a teacher).

Additionally, within the *Skills* section of the section level, self-assessment labels are extracted for degrees of competence of individual skills. Using this data, all skills found within the CV can then be sorted according to the following criteria: competence by self-assessment, duration of use, time since last use, and importance of a skill within the employment.

B. ITEM LEVEL

As part of the item level information we have defined 13 different labels:

- **NAME:** first and last name. Example: *As a strategy consultant in 2015, Fredrik Gillberg acquired immense experience...*
- **ADR:** an address, i.e. place of residence, usually given as street name and number. Example: *Address: Visiokatu 1 33720 Tampere, Finland*
- **MAI:** an email address. Example: *E-mail (work): angel.toribio@gmail.com*
- **NMR:** a phone number. Example: *Nivico contact: +46 73 78 88 073*
- **LAN:** a spoken language. Example: *Languages: Finnish - Native, English - Fluent, Swedish - Basic*
- **LOC:** a geographical location including continents, countries, states, provinces, counties, regions, cities, villages and muncipalities. Example: *Working as Chief Software Architect at LATO Oy , Espoo, Finland since 2012 till date*
- **DAT:** an individual date containing day, month and year; month and year only; or year only. Example: *EXPERIENCE: Avalanche Studios 2013 âĂŞ 2015*
- **DUR:** a time period. Example: *Rasmus is a full stack project manager with more than ten years experience...*
- **ORG:** a name of an organization, an institution or a business subject. Example: *Consultant / Software Designer. KajaPro Oy , Oulu . Finland 01 / 2012 âĂŞ 04 / 2012*
- **ROL:** an job title or a profession. Example: *Mihail is a senior embedded software developer, currently holding...*
- **EDU:** a name of an educational institution: schools, universities, faculties and institutes. Example: *Education: 2010-2016 Aalto University School of Electrical Engineering*
- **DEG:** information about an educational degree, usually a diploma or academic title. Example: *I obtained a masterâĂŹs degree in statistics (subtopic: machine learning)...*
- **CER:** a course certificate. Example: *CSA Sun Certified System Administrator for Solaris*
- **O:** denotes a token not belonging to any of the above categories.

C. SECTION LEVEL

As part of the section level information we have defined six different labels:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

- **Personal:** personal information such as first and last name, address, phone number, email, location of residence, nationality, gender, date of birth, and language. Example: *Angela Mikkonen, S- 165 71 HÃd'sselby, Mobile: +46 70 830 0929, e-mail: lasc@valcon.se*

- **Emp_INFO:** basic information about a single employment, usually including start and end date, name of organization, and job title. Example: *Senior management consultant, Valcon (2010-2015)*
- **Emp_DESC:** a description of the task within an employment, usually after the *Emp_INFO* section. Example: *Identified critical areas and issues to improve / Driver diagrams. Established cross-functional process and process organization*
- **Education:** a description and location of primary, secondary or higher education; diplomas, certificates, and various courses. Example: *Valcon âĂIJConsulting Skills TrainingâĂİ (2011-2012), Lean, LuleÃě University of Technology (2010)*
- **Skills:** competences, areas of expertise. Example: *Business Performance management and operational performance measurement. Customer Journey Mapping and service design. Java, C++, C, Objective C*
- **O:** denotes a section not belonging to any of the above categories.

D. SELF-ASSESSMENT OF SKILL COMPETENCE

We have defined four self-assessment labels representing degrees of competence in a particular skill:

- **Excellent:** high proficiency, advanced level. Example: *Java 8 - Excellent, ARM - Good*
- **Good:** intermediate level. Example: *Level of knowledge / duration used (years): Ansible 5/5 3, Django 3/5 1*
- **Bad:** beginner level. Example: *Skill self-assessment (3 max): English ***, German *, Swedish **
- **Null:** self-assessment is missing. Example: *Skills: Java, Python, Django*

E. MODEL ARCHITECTURE

For our purposes, two models were created, a dual model that extracts both section and item level information from a CV document, and a model for self-assessment of skill competence which receives the extracted **Skills** section from the dual model and performs classification of its content.

Both models consist of a multilingual pre-trained BERT model [4], that provides a contextualized vector representation for each input token, and two linear layers (one for the section level, the other for the item level of the dual model) that individually receive the vector representations and perform classifications. In the case of a skill assessment model, two linear layers are still used regardless of the fact that the model predicts labels on a single level. One layer is a standard classifier with the number of dimensions equal to the number of different competence degree labels, while the other layer has one-dimensional output with the sigmoid activation function, giving the floating point output from $[0, 1]$. If the true label is not **Null**, the loss of the second linear layer is calculated as the mean squared error (MSE) and is added to the standard cross-entropy loss of the first layer. The second layer is only used during training, to provide information about how close the prediction is to the actual label (e.g. the prediction **Excellent** is closer to **Good** than to **Bad**). The MSE part of the loss is defined as $loss_{MSE} = \left(\frac{label_{real} - 1}{3} - pred_{score}\right)^2$, where $label_{real}$ is the actual competence degree (from 1 to 4 for **Bad, Good, Excellent**), and $pred_{score}$ is the layer output with sigmoid activation function.

Raw CV text is converted into tokens using a WordPiece tokenizer and an *item_index* vector is introduced in which *True* denotes the first token of each word, and *False* denotes other tokens, so that individual words can be broken into multiple tokens. Using the index we ensure that each word of a CV will have only one final label of the corresponding level at the output of the model – the one corresponding to the beginning of that word.

This representation is actualized in the initial part of the dual model, the multilingual BERT and through the linear *item* layer, after which the *True* indices of the *item_index* vector are taken and for them, in learning phase, backpropagation is performed through the whole model, i.e. they are returned during usage phase. For the *section* level input, the same WordPiece tokenization is used and an additional **[NEW_LINE]** is introduced to mark the beginning of each CV line. Since different sections of a CV usually start and end on lines within the raw text, in this way the formatting information is preserved (which is often lost because some sections do not use punctuation at the end of sentences). In addition to the new token, a new *section_index* vector is introduced with values set to *True* for each **[NEW_LINE]** token, and *False* for all other tokens. This model's behaviour is depicted in Fig. 1.

This allows the section part of the dual model to perform the classification for each line regardless of its variable length in a single pass, and also allows this mode of operation with mini-batches. Although section classification is performed for one token per line only, contextualization by the left and right contexts using the transformer allows each of these tokens to have a completely different (and separable using the later fully connected layer) vector representation. Since the section part of the model works at the line level, variability of the model output is reduced compared to the model that would provide a classification for each token, which is important for the section level where individual sections may contain hundreds of tokens.

Backpropagation in the learning phase is performed only for tokens marked with *True* within *section_index* vectors, i.e. only for **[NEW_LINE]** tokens, while during the usage phase each marked line is expanded by the number of words in that line, which gives a unique section classification for

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

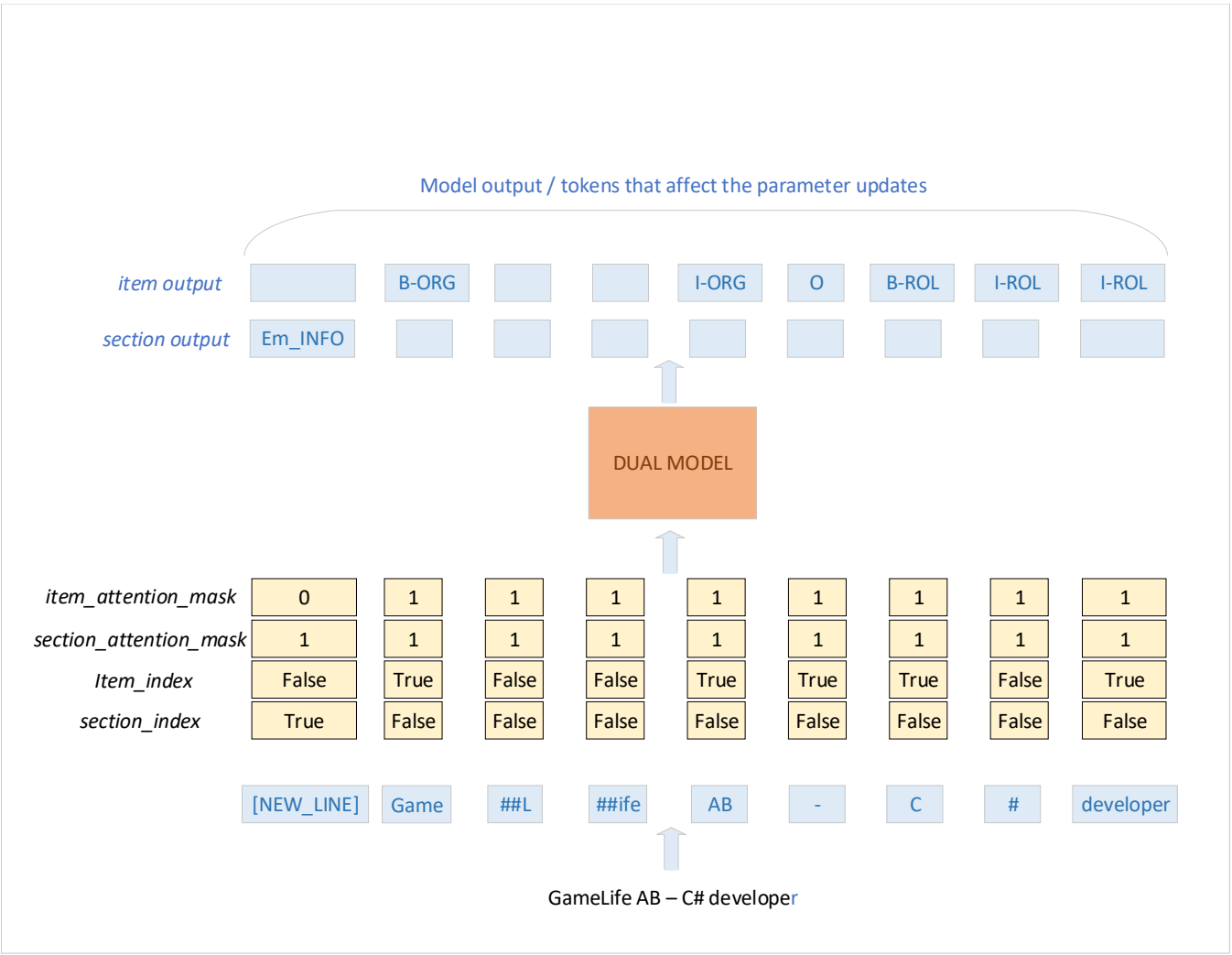


FIGURE 1: Dual model: example of one pass.

each word in the CV with the restriction that all words within the same source line have the same classification. In order to (respectively) disable/enable the visibility of **[NEW_LINE]** tokens in the item/section level pass of the model, two new vectors are introduced: *item_attention_mask* and *section_attention_mask* by which (using the Hadamard product) the attention vectors within BERT are multiplied. Thus, the desired tokens are hidden, such as the token for a new line in the item model pass, and the padding token **[PAD]** for both passes of the model where it is required to achieve equal sequence lengths within the same mini-batch.

For the skill competence assessment model, along with the initial tokenization, an additional **[NEW_LINE]** token is used (as in the section part of the dual model) and a new **[SKILL]** token is used to replace all tokens belonging to a skill. The replacement enables a unique label for any required skill, giving the model accurate information about whether the currently viewed token is a skill or not, without the need to infer the same information using the context of the token. *skills_index* is also introduced, which indicates positions of the skill tokens, i.e. positions for which the classification is determined. Since the skill names for which the assessment is performed are obtained from an external database, positions denoting the skills are obtained by simple intersection with the skills from the database. The database contains 24 403 skill names for various professions scraped from LinkedIn. All skills have been translated into five respective languages. Therefore, the task of the model is to infer the skill self-assessment (and whether it exists) exclusively from context, rather than finding the positions of all skills within a CV. An example of a single pass through this model is depicted in Fig. 2.

Fig. 3 presents the complete system architecture during the usage phase. It contains the following modules: input – a free-form CV, parsers for supported document formats, WordPieces tokenizer, pre-processsing modules, a module for post-processsing the output of the section level and the skill model, a *hard-match* skill extractor, the dual model and the skill competence self-assessment model.

The input to the system starts with a given CV docu-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

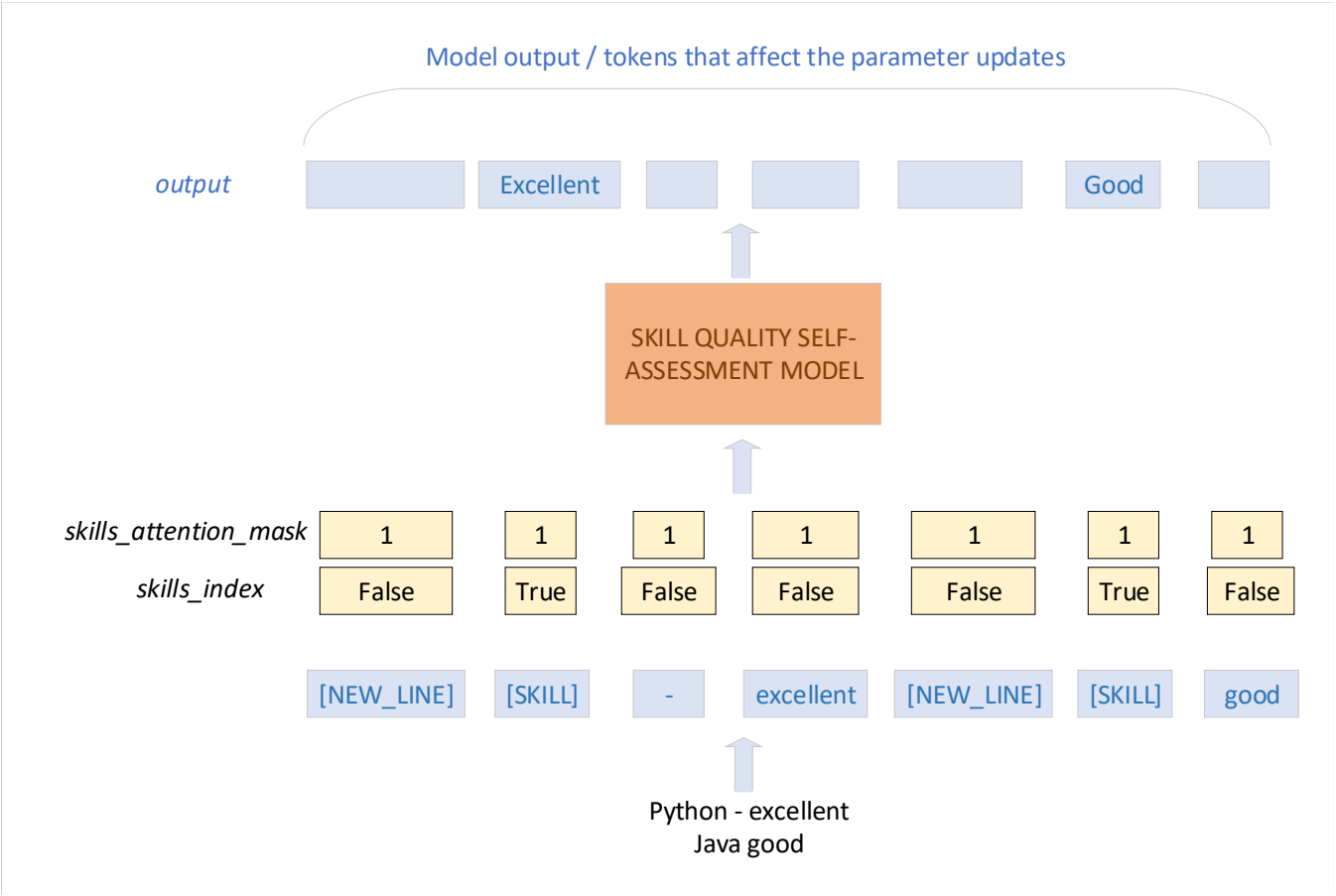


FIGURE 2: Skill assessment model: example of one pass.

ment for which useful information is to be extracted. The CV can be in any of the following formats: txt, rtf, doc, docx, odt, or pdf. The document goes through a parser module that extracts the raw text from the submitted file. The raw text then passes through a WordPieces tokenizer that separates the input into tokens that the models recognize. The tokens then go through the section pre-processing module which adds **[NEW_LINE]** tokens before the first token of each line of the original CV. *section_index* and *section_attention_mask* vectors are also created. Similarly, the input to the item pass of the dual model is obtained by passing through the item pre-processing module where *item_index* and *item_attention_mask* vectors are formed.

The inputs pass through the dual model which returns the corresponding class for each **[NEW_LINE]** and for each unmasked item token. The section level classification additionally goes through the output post-processing module, which expands the classification of each line to each token belonging to that line, in order to obtain a section class for each item level token. This is the first part of the system output.

The second part of the output starts with selecting tokens that belong to the **Skills** section, which is obtained by the dual model pass. The raw CV text further goes through a hard-match skill extractor module that gets the skill names from an external database of the required skills and simply searches for the corresponding strings in the CV. Positions of all tokens that represent skills and belong to **Skills** sections are replaced with the **[SKILL]** token and, inside the pre-processing module, *skills_index* vector is created.

Such an input goes through the model for self-assessment of the skill competence degree that for each skill (i.e. each **[SKILL]** token returns the estimated competence degree. The model output goes through a post-processing module that returns a list of skills with their classifications, which is the second part of the system output.

F. MODEL TRAINING

Both models receive input with a length of 384 tokens. No classification is performed over the first 128 tokens; they serve as an extended context for classifying the other 256 tokens, which is enabled by setting the first 128 positions of individual *MODEL_index* vectors to *False*. For initial CV tokens where it is not possible to have an extended context, classification is performed over all tokens.

Mini-batches of 16 examples are used. Loss function in the dual model is defined as the sum of cross entropy losses for prediction in section and item levels. In the skill self-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

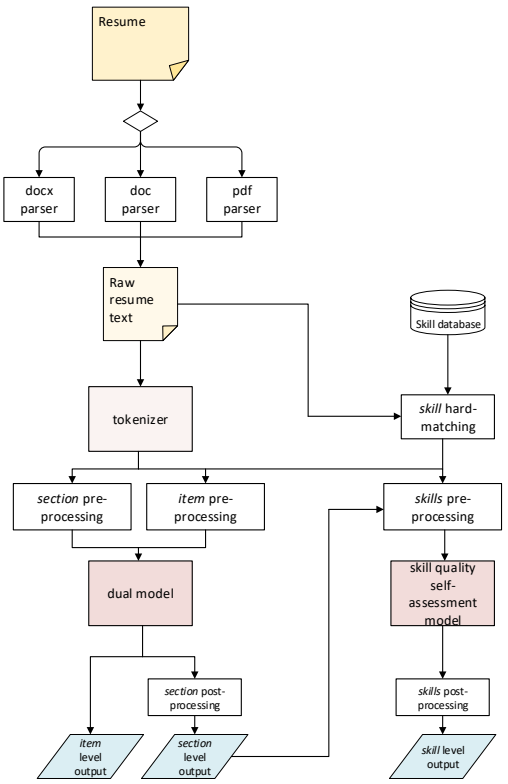


FIGURE 3: High-level system overview

assessment model, the loss function is defined as the sum of the cross entropy and the mean squared error for the skills not marked with **Null**, i.e. having an associated self-assessment. Addition of the mean squared error loss allows for obtaining additional information during learning about the proximity of the prediction to the correct label.

The models were trained until the error on the validation set stagnated or decreased through three consecutive epochs, or five epochs in the skill assessment model. We use the the AdamW (*Adam with decoupled weight decay (5)*) optimizer applied to all model parameters with the learning rate set to $2 \cdot 10^{-5}$ and the linear heating parameter set to 0.1. The weight decay factor used in the AdamW optimizer is set to 0.01 for all model parameters except the biases and the gamma and beta parameters of the normalization layers (1).

IV. EVALUATION

The dataset for learning the classification model for item and section levels of a CV consists of 1686 annotated CVs in five languages: English, Swedish, Norwegian, Finnish, and Polish. Of these, 762 are in English, 242 in Swedish, 188 in Norwegian, 275 in in Finnish and 255 in Polish. The dataset for learning the skill self-assessment model consists of 714 annotated CVs in English.

In annotation of the dataset for item level and skill assessment, **BIO** notation scheme was used, where the initial token of a phrase is marked as *B-Category* and other, internal tokens are marked as *I-Category*. (Example: *software engineer* is marked as **B-ROL** and **I-ROL**.) This way the annotation preserves information about the beginning and end of a multi-token phrase. Each section level token is simply marked as its own category, without B or I prefixes, because at this level the information of the beginning or the end of a sequence is not important: the level is used only as a contextualization of the item level.

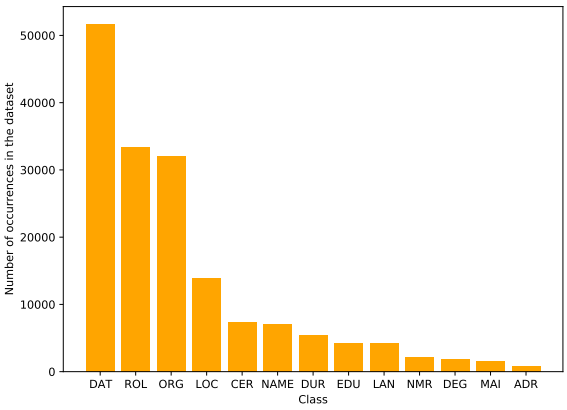


FIGURE 4: Item level: category frequencies in the annotated set (excluding **O** which appears 1 613 123 times).

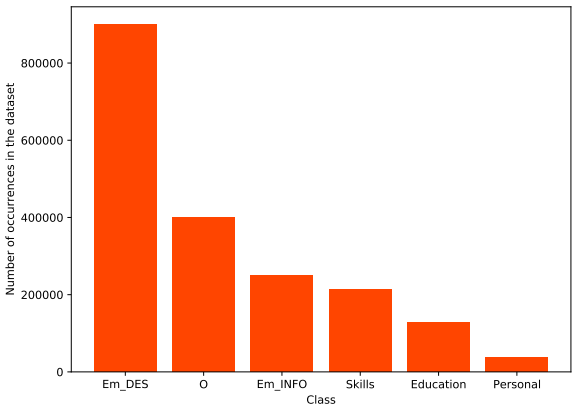


FIGURE 5: Section level: category frequencies in the annotated set.

Labels within the dataset are not numerically balanced and there are large differences in number of individual categories. This imbalance is shown in Figs. 4 and 5.

The dataset used to learn the model for self-assessment of the skill competence degree was annotated in a different way. Here, the names of all required skills were obtained from the external database, and their positions were found in the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

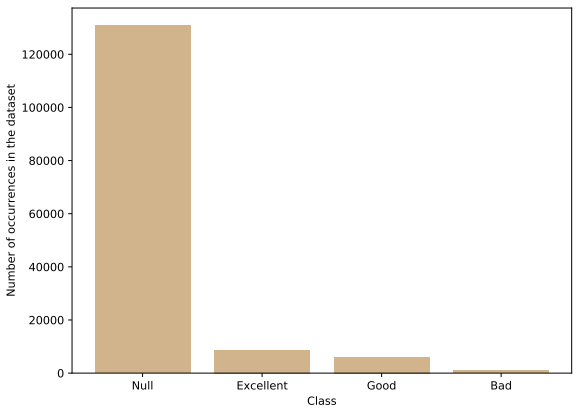


FIGURE 6: Skill assessment: category frequencies in the annotated set.

CV within the **Skills** section by simple string matching. The model in this case does not have to specify positions of the skills themselves, but to make a decision as to whether there is a competence degree in their context. Therefore, instead of the **BIO** scheme, only four classes defined in Sect. III-D were used. This set also shows a large imbalance in the number of examples of each class, shown in Fig. 6.

The models described in the previous section are implemented within the Pytorch software framework, while for the pre-trained BERT model a *transformers* library is used ([20]). Learning was performed on NVIDIA TITAN RTX graphics card using CUDA version 10.1. A 5-fold evaluation was performed and the results are presented in the following subsections.

A. RESULTS FOR DUAL MODEL

Table 1 shows precision, recall and F1 measure for each section level class obtained by 5-fold evaluation on the dataset. The model shows best performance on **Education** and **Em_DES** classes, while the worst performance is shown on **O** parts of the CV which could be explained by wide range of content that can be found in this section, which is more thoroughly investigated in Sect. V-B. In addition, Table 2 shows macro-averaged measures of the dual model at the section level.

Since one of the main requirements of this model was multilingualism, below we show the results for each individual language. The model shows best results at section level for Swedish and Polish despite having fewer CVs, which can be explained by the BERT characteristic to learn relevant representations in other languages even though the majority of the dataset was in English ([21]). Results for each *section* level class per language can be found in A.

Table 4 shows precision, recall and F1 measures for each of the item level classes obtained by 5-fold evaluation on the dataset. Here the best results are obtained on **O** tokens which were several times more numerous than other tokens due to the dataset imbalance, which is the likely reason for giving greater capacity for this class to the model. Of the other classes, the best results are obtained for those with the simpler content, such as **DAT, LAN** and **MAI** classes. The worst performance is shown on **CER** because this class usually appears without a context, as a list item along other education-related terms. This fact gives little information to the model that the token belongs to **CER** class. Also, in the learning set, there are few such tokens in comparison to others (4.4% of tokens excluding **O**). Table 5 shows the macro-averaged measures of the dual model at the item level.

As for the section level, the results for each item are shown at the item level for each language. Here, as expected, the model achieves the best results for English, for which it also had the most examples in the learning set. The results for each *item* level class per language can be found in B.

Figs. 7 and 8 show visualizations of the dual model output on the test example. While section level is perfectly labeled in this example, item level has small errors that should be manually cleaned up later, such as labeling a bracket after the organization name as part of that name.

In addition, dual models were trained where the last fully connected layer received concatenated hidden representations of the last two or three layers of the BERT model so that the classifier had different levels of information available when making predictions. Both versions of the model were trained with a 5-fold evaluation. With a significance level of $p = 0.05$, no statistically significant increase in precision, recall, or F1 measure was observed in comparison to the original dual model, whose last linear layer only received representations from the last layer of BERT.

Fig. 9 shows the effect of changing the number of BERT layers, i.e. different model sizes, on F1 measure at the validation set. We can see that the higher number of layers before the final linear layer improves the model performance, but each additional layer achieves less and less improvement on the final performance. 8 layers are enough to achieve the F1 measure of 99.1% for section level and 99.3% for item level compared to the original 12 layers of the BERT model, which opens up a possibility of a drastic reduction in the model size and memory consumption along with speeding up the model, paying a small price in output quality.

Figs. 10 and 11 show the ratios of F1 measure to the best F1 measure of each class for section and item level (respectively) with respect to different numbers of BERT layers. The section level shows a monotonous increase in performance up to the eighth layer of BERT where they remain roughly constant until the last (12th) layer, indicating that increasing the model would not positively affect the F1 measure at the section level. On the other hand, changing the number of layers has a wide range of effects on performance at individual item level classes. The largest positive impact is seen on the **CER** class, whose ratio increases almost linearly as the number of layers increases, indicating that adding new BERT layers could further improve the performance for this

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

	Personal	Education	Em_INFO	Em_DES	Skills	O
Precision	0.82176	0.91452	0.8772	0.88662	0.82464	0.76326
Recall	0.79824	0.8954	0.85475	0.90089	0.7866	0.79618
F1	0.80938	0.90459	0.86534	0.89341	0.80427	0.77865

TABLE 1: Section level of the dual model: 5-fold evaluation results.

Consultant Profile
Michael Smith
E-mail : michael.smith@hotmail.com
Tel : +46 99 65 62 575
Home address : SkolspÃ¥ret 73 , 620 10 BURGSVIK , Sweden Personal
Previous work
2016 - 2020 InfoGames AB , Team leader Em_INFO
I lead a team of 6 , developing a 3D platformer game
nominated for the game of the year 2019. Em_DES
Skills : Unity3D , Blender , C# Skills
2010 - 2016 SquareSpace , Full stack developer Em_INFO
Worked on both front and back-end to provide scalable solutions.
I was responsible for the full development cycle. Em_DES
Skills : Java, JavaScript, HTML, PHP, C# Skills
2005 - 2010 SWERouting AB , Junior Java developer Em_INFO
I worked as a junor developer working on routing analysis and route storage. Em_DES
Skills : Java
Education
1994 - 2000 Bachelor of Applied Physics National University of Stockholm , Sweden
1991 - 1994 School graduate with an increased level of physics and mathematics, Swedish Physics-Mathematics
Liceum of National University of Stockholm , Sweden
Microsoft Certified IT Professional: Enterprise Admin (Microsoft)
Citrix Certified Administrator for PS4 (Citrix) Education
Skills level / years of use
Java - 5 15 years
JavaScript - 5 15 years
HTML - 5 20 years
Unity3D - 5 8 years
C# - 5 8 years
Python - 3 2 years
Numpy - 3 2 years Skills
Swedish - native
English - profficient
German - basic
Personal

FIGURE 7: Dual model output example (section level).

class. This is not true for every class, for example **ADR** and **MAI** start to lose on the F1 measure in the later layers of the model, which opens the possibility of setting the final classifier on different layers for different classes, so that prediction is performed with the optimal number of layers for that class.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

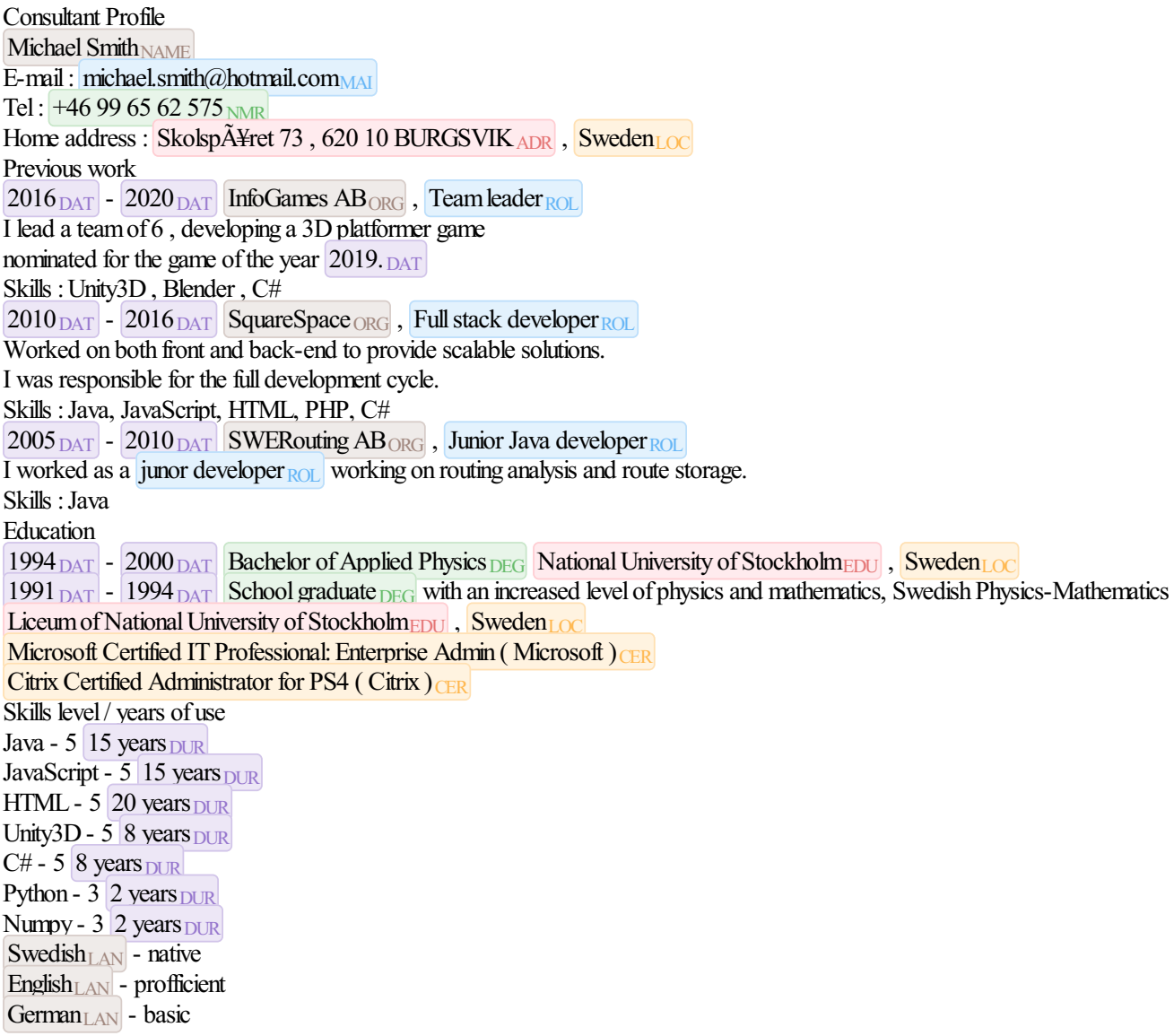


FIGURE 8: Dual model output example (item level).

	macro-average
Precision	0.83356
Recall	0.83732
F1	0.8344

TABLE 2: Section level of the dual model: 5-fold evaluation results (macro-averaged).

	Norwegian	Swedish	Finnish	Polish	English
Precision	0.863	0.8811	0.86792	0.88172	0.82821
Recall	0.85209	0.87223	0.85297	0.86327	0.81713
F1	0.85578	0.87587	0.85686	0.87046	0.82175

TABLE 3: Section level of the dual model: 5-fold evaluation results (macro-average, all languages).

B. RESULTS FOR SKILL SELF-ASSESSMENT MODEL

Table 7 shows the 5-fold evaluation results for the skill assessment model on the annotated set. The best performance is obtained on the **Null** class, while the measures for other classes are lower. There are two reasons for the weaker performance of this model compared to the dual model. First, there is again a major imbalance between classes and the model again works best for the class that was most numerous in the learning set.

Another reason is incorrect extraction of raw text from various supported formats using parsers. Data on skills and their qualities are often found in tables, which are difficult to copy from a format that supports them (e.g. docx or pdf) to a txt format used as input to this model. The problem occurs when the information is extracted from the table in the wrong order. For example, if the table consists of two rows, where the first row lists the skills and the second row gives their competence assessments, the parser will first extract the list

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

	ADR	CER	DAT	DEG	DUR	EDU	LOC
Precision	0.69412	0.68553	0.94232	0.72963	0.72349	0.74765	0.78362
Recall	0.80416	0.7481	0.97547	0.71022	0.78997	0.76873	0.86648
F1	0.74432	0.71322	0.95857	0.71911	0.75451	0.75691	0.82256
	LAN	ORG	MAI	NMR	ROL	NAME	O
Precision	0.88444	0.75333	0.78967	0.70527	0.8141	0.75438	0.97512
Recall	0.930655	0.8472	0.94562	0.9555	0.77033	0.8875	0.96275
F1	0.90671	0.7973	0.86057	0.80915	0.79104	0.81315	0.96889

TABLE 4: Item level of the dual model: 5-fold evaluation results.

	macro-average
Precision	0.81156
Recall	0.84514
F1	0.82579

TABLE 5: Item level of the dual model: 5-fold evaluation results (macro-average).

	Norwegian	Swedish	Finnish	Polish	English
Precision	0.81695	0.84647	0.84846	0.85737	0.8576
Recall	0.71478	0.77073	0.76342	0.7804	0.80385
F1	0.7536	0.8019	0.79636	0.81048	0.82663

TABLE 6: Item level of the dual model: 5-fold evaluation results (macro-average, all languages).

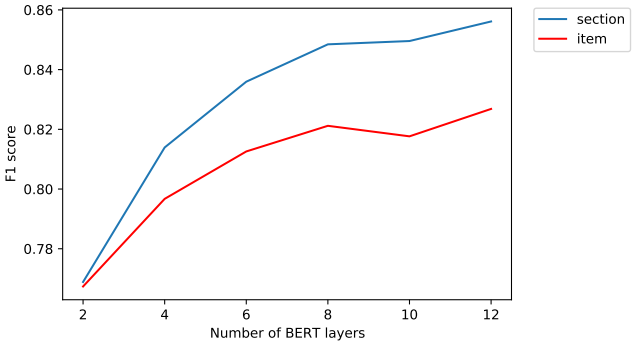


FIGURE 9: Influence of the number of BERT layers before the last linear layer to the macro-average F1 scores of the dual model.

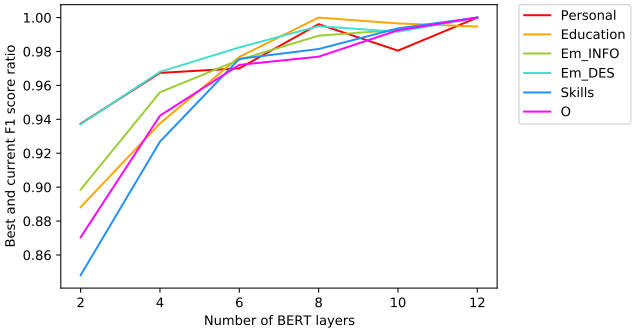


FIGURE 10: Influence of the number of BERT layers before the last linear layer to the macro-average F1 scores of the individual classes of the dual model (section level).

of all skills and then the list of qualities. The dual model will

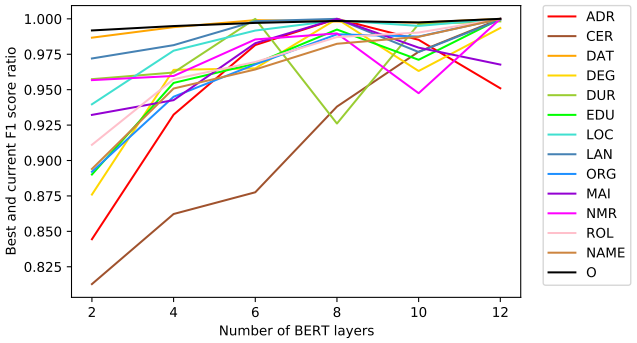


FIGURE 11: Influence of the number of BERT layers before the last linear layer to the macro-average F1 scores of the individual classes of the dual model (item level).

	Null	Bad	Good	Excellent
Precision	0.88269	0.51351	0.55816	0.46386
Recall	0.83938	0.56436	0.49798	0.61011
F1	0.86049	0.53774	0.52635	0.52703

TABLE 7: Skill assessment model: 5-fold evaluation results.

Skills : Unity3D Null , Blender Null , C# Null
Skills : Java, Null JavaScript, Null HTML, Null PHP, Null C# Null
Skills : Java Null
Java Null - 5 15 years
JavaScript Excellent - 5 15 years
HTML Excellent - 5 20 years
Unity3D Excellent - 5 8 years
C# Excellent - 5 8 years
Python Good - 3 2 years
Numpy Good - 3 2 years

FIGURE 12: Output example of a skill assessment model on **Skills** level of the test example (obtained by the dual model output)

.

label the first part as **Skills**, while the list of qualities will be labeled as **O** so the skill assessment model will not even receive information on the quality of individual skills. If the table is extracted in a semantically correct way, as shown in Fig. 12, the model achieves good performance, both in cases where qualities are described by words such as *decent knowledge, excellent, long term user*, and in cases where the assessments are presented by symbols or numerically.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

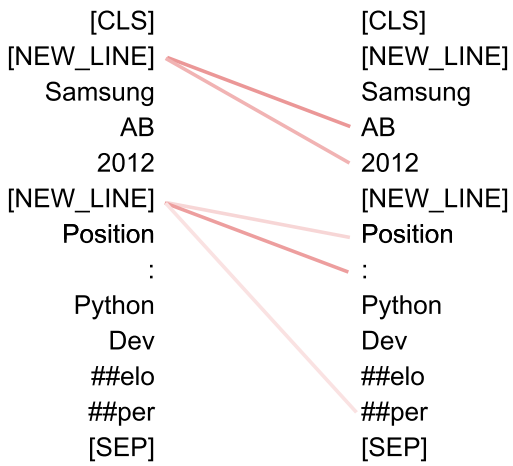
	macro-average
Precision	0.60456
Recall	0.628
F1	0.61603

TABLE 8: Skill assessment model: 5-fold evaluation results (macro-averages).

Similary to the dual model, we performed a 5-fold evaluation of the model which, apart from the last hidden representation, receives the representations of the last 2 or 3 layers of the BERT model. Again, we find no statistically significant improvement for $p = 0.05$.

V. MODEL EXPLAINABILITY

A. VISUALIZATION OF ATTENTION OF THE TRAINED MODELS



(a) Attention of the *[NEW_LINE]* token, head 4 layer 6.

(b) Attention of the *[NEW_LINE]* token, head 4 layer 7.

FIGURE 13: Different role of the *[NEW_LINE]* token inside different heads of the dual model. Visualization obtained using [22].

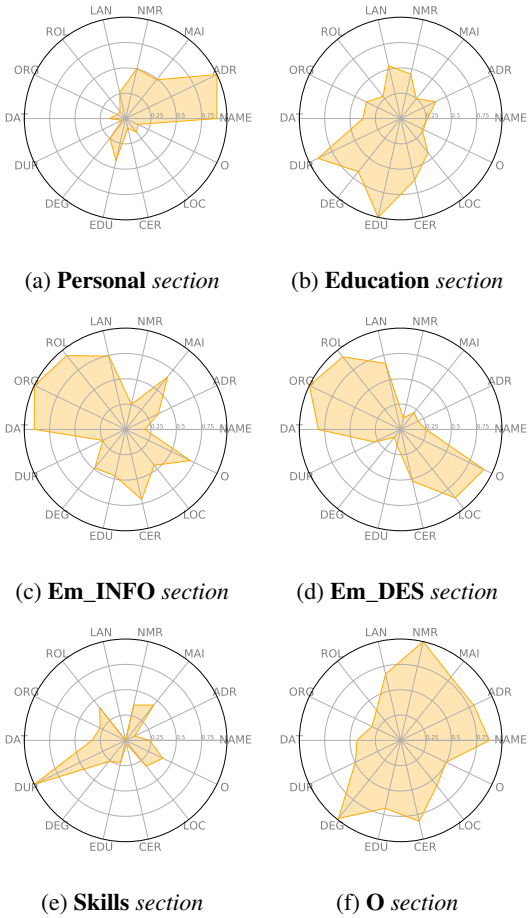
The attention function used inside the transformer allows the model to be interpretable. Following the example of [23], attention was investigated on different layers of the model. The research was particularly focused on newly introduced tokens to confirm that the model uses them in the way they were designed and that their introduction is not redundant. Darker lines on the visualizations indicate stronger attention between the connected tokens. In the following examples a wide range we illustrate the wide range of attention patterns within the individual heads of the model, we observe simple patterns such as those that pay attention to the next token or the previous token, to more complex heads that have wide attention for individual special tokens like **[CLS]** and **[NEW_LINE]** tokens that are trained to contain information about the whole input or the entire line. An example of this type of head is presented in more detail in Fig. 13 where different roles of the same token can be observed within two different attention heads on different layers. In Fig. 13a the special newline token has attention focused on the line to which it belongs, while within the second head in Fig. 13b its attention is on the previous line, indicating the variety of information the model may contain in only one token.

Furthermore, the interaction of item and section levels when predicting section levels is investigated. For each section class, we counted the token labels of the item level which had the highest amount of attention within each head of the dual model looking at the **[NEW_LINE]** token. Specifically, for each layer of the model a tensor was obtained with dimensions $12 \times num\_section\_tokens \times num\_item\_tokens$ which contains the attention values for each **[NEW_LINE]** token to each item token. Then the index with the highest attention value was found for each **[NEW_LINE]** token – the position of the item level token which is most important for the prediction of the current section level. Classes of these tokens are counted for each level and head of the model. The values were further normalized by dividing by the total number of counted maximal tokens of the corresponding class in order to avoid distorted assignment of importance to the more numerous token class due to an unbalanced input set. Also, all values within a section class are divided by the maximum value within that class to keep all values in the range of zero to one. The results are shown in Fig. 14.

From the diagrams shown, it can be seen that the dual model has successfully learned the importance of certain item level classes with respect to the section level and gave them most attention when predicting. For the prediction of the **Personal** section (Fig. 14a), the model pays most attention to the **NAME**, **ADR**, **MAI**, and **NMR** tokens, which correspond to the most common forms of personal information in CVs. Most attention of the **Education** section (Fig. 14b) is given to tokens **DEG** (level of education), **EDU** (organization where the level or certificate was achieved), and **CER** (type of certificate). High attention is also paid to **DUR** tokens here because this section often mentions the length of study. **Em_INFO** (Fig. 14c) and **Em_DES** (Fig. 14d) have similar patterns of attention, attaching importance to employment-related tokens (**ROL, ORG**) and the beginning

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

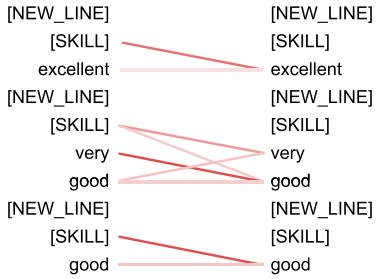


(a) **Personal** *section* (b) **Education** *section*

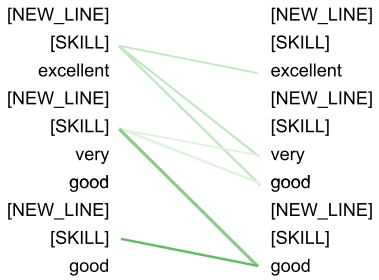
(c) **Em_INFO** *section* (d) **Em_DES** *section*

(e) **Skills** *section* (f) **O** *section*

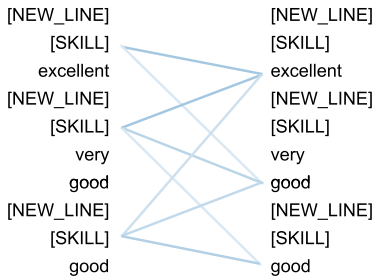
FIGURE 14: Item level classes with the largest attention values inside each head of all layers of the dual model, counted for all **[NEW_LINE]** tokens during section level prediction.



(a) Attention of the **[SKILL]** token, head 4 layer 2.



(b) Attention of the **[SKILL]** token, head 3 layer 5.



(c) Attention of the **[SKILL]** token, head 1 layer 5.

FIGURE 15: Different roles of the **[SKILL]** tokens inside different skill assessment models. Visualization obtained using [22].

and end of employment (**DAT**). The **Em_INFO** section also pays more attention to education-related tokens, which could be explained by the similarity of the structures of these two sections within the CV which forces the dual model to pay more attention to tokens of both types in order to distinguish between **Education** and **Em_INFO**. **Em_DES** uses more attention on **O** tokens for the similar reason, to successfully distinguish the **O** section of a similar structure. Since the **Skills** section (Fig. 14e) inside the item level has no special label to indicate the location of this section, the model pays attention to the only relevant data that is often found next to the skills: **DUR** token (length of usage). Section **O** (Fig. 14f) also has no clear indicator of its class so it has wide attention, monitors relevant tokens of other classes, and performs prediction by elimination: if these tokens are not present, it is an **O** section.

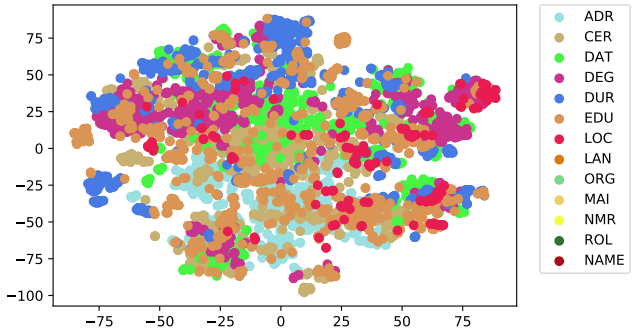
Like the **[NEW_LINE]** token in the dual model, the **[SKILL]** token in the skill assessment model has a different function within different model heads when providing information during classification. In earlier layers, such as layer 2, head 4 shown in Fig. 15a, the interaction of the **[SKILL]** token with other tokens is simpler, attention is focused on tokens that assess quality for the current skill. In the later layers, more complex interactions begin to appear, such as the one in in head 3 of layer 5 where the attention of the token is (in addition to its own quality) focused on tokens describing the qualities of the **[SKILL]** tokens that appear after the current token. Within head 1 of layer 5, attention is widely distributed to all quality labels within the input sequence.

B. VECTOR REPRESENTATIONS OF THE TRAINED MODELS

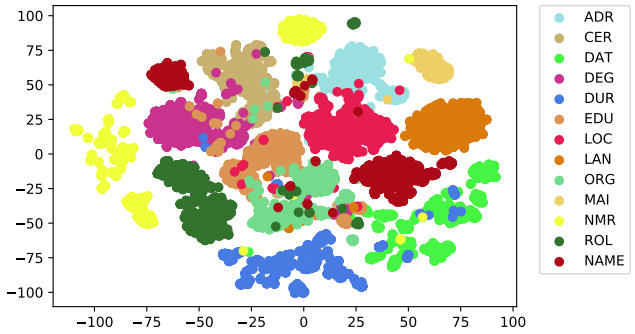
We now investigate the change in the contextualized vector representations of the tokens from the pre-trained multilingual BERT model (which was the basis of the dual model)

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access



(a) Pretrained dual model



(b) Fine tuned dual model

FIGURE 16: Vector representations of the pretrained and fine tuned dual model for item level tokens inside the validation set for each class, reduced to two dimensions using t-SNE algorithm.

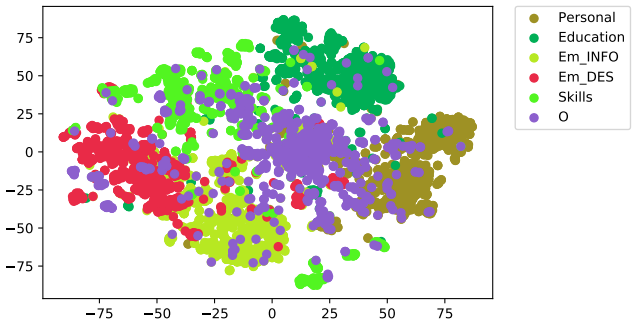


FIGURE 17: Vector representations of the fine tuned dual model **[NEW_LINE]** tokens of the section level inside the validation set for each class, reduced to two dimensions using t-SNE algorithm.

to the fine-tuned dual model. Fig. 16 shows these item level representations before and after learning the model. The dimensionality of the vector representations was reduced to 2 using the t-SNE [24] algorithm. Before learning, clusters can be seen for each class, indicating the similarity of representations of individual tokens which are close in this space, but most clusters have a good portion of scattered tokens, whose representations overflow the boundaries of other clusters. On the other hand, the clusters of the trained model are less dispersed and the vast majority of examples is within the clear boundaries of their class with a certain distance from other clusters, which allows their separation and final classification by the next linear layer of the model. A similar pattern is observed in Fig. 17 which shows clear boundaries of clusters defining individual classes, except for class **O** whose cluster is exceptionally scattered. This dispersion is due to a high variation in the content itself that can be found in this section, as well as the context that may surround the **O** section.

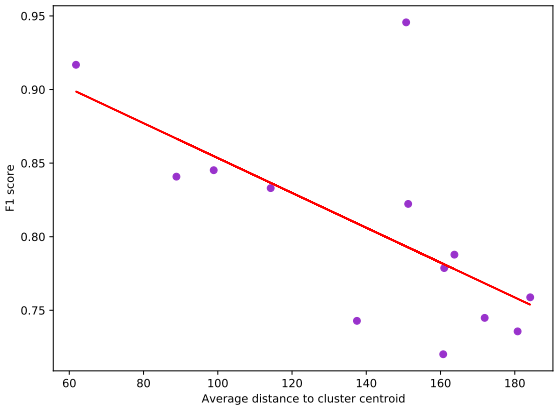


FIGURE 18: Correlation between the average distance from cluster centroids and F1 measure for each item class on the dual model validation set.

The correlation between dispersion (and size) of the clusters and the performance quality on the validation set for certain item classes are shown in Fig. 18. For each cluster in Fig. 16b the centroid was calculated, then for each example within the cluster its distance from the centroid was calculated and the distances were averaged. It can be seen from the figure that there is a negative correlation between dispersion of the cluster and the F1 measure of the dual model (Pearson's correlation coefficient equals -0.64352). The upper right corner shows an example of a class that deviates from the negative correlation: the **DAT** class, which can be found in a large number of contexts within the CV, so the vector representations of dates can be very different. But since a date is still usually presented as a single number denoting the year, the dual model maintains a large F1 measure over this cluster despite a more dispersed representation.

Fig. 19 shows the vector representations of all **[SKILL]** tokens within the validation set of the model that achieved the best performance on its fold. It can be seen that although there are distinct clusters for each class, the clusters are quite close to each other and there is a large area of overlap between them. This indicates a weaker performance of this model compared to the dual model because during learning it failed to fully tune the representations to make separation into different classes possible.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

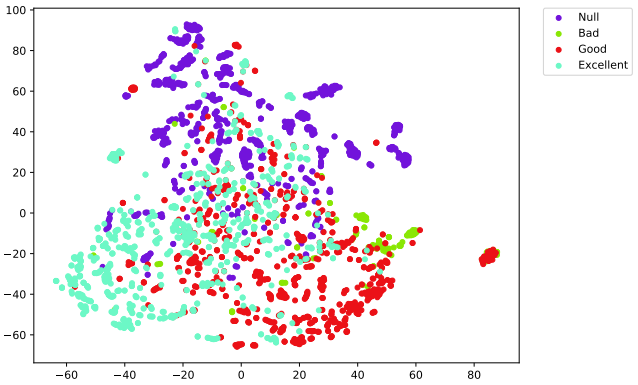


FIGURE 19: Vector representations of the fine tuned dual model **[SKILL]** tokens of the section level inside the validation set for each class, reduced to two dimensions using t-SNE algorithm.

VI. CONCLUSION

This paper proposed a new architecture for processing sequential inputs using transformer, and the implementation of its encoder part in the form of the BERT language model. BERT was used as a basis for construction of two multilingual models for the extraction of useful information from free-form CVs, and both were tested on the annotated datasets for the two problems.

The first model was used for a dual purpose: extracting "hard" information such as names or previous employment organizations, and contextualizing the extracted information by classifying individual parts of the CV into sections. The dual model on this problem achieves a macro-averaged F1 measure of 0.8334 at the section level and 0.82579 at the item level. Performance is only slightly lower for languages for which far fewer examples have been annotated compared to the English language. The model achieves the F1 score of 0.86, 0.88, 0.86, 0.87, and 0.82 on section level, and 0.75, 0.80, 0.80, 0.81 and 0.83 on item level for Norwegian, Swedish, Finnish, Polish, and English, respectively. By investigating the impact of the number of BERT layers before the final classification layer we observed that using only 8 layers achieves over 99% of F1 measures at both levels of this model. We note the possibility of setting classifiers on different levels of the model, since some classes are more easily separable in earlier layers of the model.

The second model was used to detect self-assessed skill competence degree, where for each skill found in a CV the model determines whether there is an associated quality. On the corresponding dataset, the model set achieves an F1 score of 0.61603 despite the lack of examples of a particular class and incorrect parsing of tabulated data into raw text.

We also focused on interpretability by visualizing the attention of the model. It is shown how the newly introduced tokens **[NEW_LINE]** and **[SKILL]** have acquired the expected functionality by learning. The attention of the **[NEW_LINE]** tokens of individual section classes corresponds to semantically related item classes, and attention is focused on current line tokens or, in later layers, on other input lines to collect more information before final classification. Similarly, **[SKILL]** token in early layers pays attention to tokens directly related to the descriptions of the currently observed skill, while in later layers attention is also directed to the wider context. Vector representations of different classes were also visualized and a negative correlation was observed between the dispersion of a cluster of representations and model performance in the corresponding class.

In future work, the performance of the model could be improved by annotating additional CVs that are focused on the classes that are least common in the current dataset. Also, expanding the dataset with more CVs in other languages would increase the performance achieved in those languages. Additionally, the property of multilingual BERT to generalize changes in parameters using one language to other languages opens up the possibility of testing this system on other, unseen languages, especially those with a sentence structure similar to the languages in the current learning set. A model for self-assessment of the skill competence degree would also benefit from the expansion of the learning set, and parsers of supported formats that would correctly map tabular data to raw text should be explored.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, (Red Hook, NY, USA), pp. 6000–6010, Curran Associates Inc., 2017.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," CoRR, vol. abs/1412.3555, 2014.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," CoRR, vol. abs/1810.04805, 10 2018.

[5] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 09 2016.

[6] K. Yu, G. Guan, and M. Zhou, "Resume information extraction with cascaded hybrid model," in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), (Ann Arbor, Michigan), pp. 499–506, Association for Computational Linguistics, June 2005.

[7] R. L. Stratonovich, "Conditional markov processes," in Non-linear transformations of stochastic processes, pp. 156–178, Elsevier, 1965.

[8] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, vol. 28, pp. 11–21, 1972.

[9] C. H. Ayishathahira, C. Sreejith, and C. Raseek, "Combination of neural networks and conditional random fields for efficient resume parsing," in 2018 International CET Conference on Control, Communication, and Computing (IC4), pp. 388–393, 2018.

[10] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in Competition and cooperation in neural nets, pp. 267–285, Springer, 1982.

[11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data,"

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

in Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, (San Francisco, CA, USA), pp. 282–289, Morgan Kaufmann Publishers Inc., 2001.

[12] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

[13] M. Tosik, C. Lygteskov Hansen, G. Goossen, and M. Rotaru, "Word embeddings vs word types for sequence labeling: the curious case of CV parsing," in Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, (Denver, Colorado), pp. 123–128, Association for Computational Linguistics, June 2015.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, pp. 3111–3119, 2013.

[15] P. Shivratri, P. Kshirsagar, R. Mishra, R. J. Damania, and N. Prabhu, "Resume parsing and standardization," International Journal of Computer Sciences and Engineering, vol. 3, no. 3, pp. 129–131, 2015.

[16] D. Chandola, A. Garg, A. Maurya, and A. Kushwaha, "Online resume parsing system using text analytics," Journal of Multi Disciplinary Engineering Technologies (JMDET), vol. 09, no. 01, 2015.

[17] A. Cernian, D. Carstoiu, and B. Martin, "Semi-automatic tool for parsing cvs and identifying candidates' abilities and competencies," DEStech Transactions on Social Science, Education and Human Science, 03 2017.

[18] J. Chen, L. Gao, and Z. Tang, "Information extraction from resume documents in pdf format," Electronic Imaging, vol. 2016, pp. 1–8, 02 2016.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in Proceedings of the fifth annual workshop on Computational learning theory, pp. 144–152, 1992.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," CoRR, vol. abs/1910.03771, 2019.

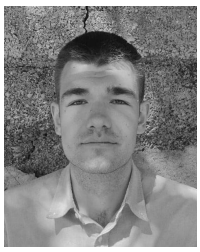
[21] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 4996–5001, Association for Computational Linguistics, July 2019.

[22] J. Vig, "A multiscale visualization of attention in the transformer model," pp. 37–42, July 2019.

[23] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bertâĂŹs attention," Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 276–286, 01 2019.

[24] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.

DAVOR VUKADIN is a research associate at the University of Zagreb, Faculty of Electrical Engineering and Computing, Consumer computing Lab. He recieved a masters degree in Computer Science from the University of Zagreb in 2020. His Ph.D. topic and research interests are related to explainable artificial intelligence in source code defect prediction.



ADRIAN SATJA KURDIJA is a research assistant at the University of Zagreb, Faculty of Electrical Engineering and Computing, Consumer Computing Lab. He received his Ph.D. in Computer Science from the University of Zagreb Faculty of Electrical Engineering and Computing in 2020. His Ph.D. project deals with service selection and QoS prediction. He has published in *IEEE Communications Letters*, *European Journal of Operational Research*, *International Journal of Web and Grid Services*, *Knowledge-based systems*, and *IEEE Transactions on Services Computing*. He is a member of the *IEEE*.

GORAN DELAČ is an associate professor at the University of Zagreb, Faculty of Electrical Engineering and Computing. He received his Ph.D. in Computer Science from the University of Zagreb Faculty of Electrical Engineering and Computing in 2014. His research interests include distributed systems, fault tolerant systems, service-oriented computing, data mining and machine learning. He is a member of the IEEE.

MARIN ŠILIĆ is an associate professor at the University of Zagreb, Faculty of Electrical Engineering and Computing. He received his Ph.D. in Computer Science from the University of Zagreb Faculty of Electrical Engineering and Computing in 2013. His research interests span machine learning, data mining, service-oriented computing, software engineering. He has published several papers in *IEEE Transactions on Services Computing*, *IEEE Transactions on Depandable and Secure Computing*, *Journal of Systems and Software*, *Knowledge-based systems*. Also, he has published his research results at the *ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* and at the *IEEE International Conference on Software Quality, Reliability and Security*. He is a member of the *IEEE*.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

.

APPENDIX A LANGUAGE EVALUATION RESULTS - SECTION LEVEL

	Personal	Education	Em_INFO	Em_DES	Skills	O
Precision	0.80869	0.91242	0.91729	0.89458	0.81756	0.82749
Recall	0.79784	0.92714	0.87924	0.89531	0.75001	0.86303
F1	0.80077	0.91941	0.89668	0.89394	0.77927	0.8446

TABLE 9: Section level of the dual model: 5-fold evaluation results (Norwegian language).

	Personal	Education	Em_INFO	Em_DES	Skills	O
Precision	0.82495	0.92422	0.91203	0.89829	0.85924	0.8679
Recall	0.76331	0.93238	0.89825	0.92495	0.85325	0.86124
F1	0.79269	0.92777	0.90451	0.91088	0.85506	0.86429

TABLE 10: Section level of the dual model: 5-fold evaluation results (Swedish language).

	Personal	Education	Em_INFO	Em_DES	Skills	O
Precision	0.89675	0.92912	0.86403	0.85441	0.86263	0.80061
Recall	0.83773	0.92052	0.79381	0.85555	0.8586	0.8516
F1	0.86543	0.92431	0.818	0.8534	0.85881	0.82121

TABLE 11: Section level of the dual model: 5-fold evaluation results (Finnish language).

	Personal	Education	Em_INFO	Em_DES	Skills	O
Precision	0.9232	0.91327	0.90092	0.89292	0.87636	0.78364
Recall	0.89373	0.89713	0.86457	0.88428	0.78617	0.85376
F1	0.90761	0.90453	0.88132	0.88792	0.82571	0.81566

TABLE 12: Section level of the dual model: 5-fold evaluation results (Polish language).

	Personal	Education	Em_INFO	Em_DES	Skills	O
Precision	0.77884	0.91953	0.8705	0.88736	0.76978	0.74323
Recall	0.75698	0.87727	0.85636	0.90708	0.73973	0.76533
F1	0.76658	0.89734	0.86328	0.89684	0.75317	0.75332

TABLE 13: Section level of the dual model: 5-fold evaluation results (English language).

APPENDIX B LANGUAGE EVALUATION RESULTS - ITEM LEVEL

	ADR	CER	DAT	DEG	DUR	EDU	LOC
Precision	0.67039	0.77863	0.98709	0.69023	0.72633	0.71872	0.72905
Recall	0.52573	0.57418	0.92473	0.72581	0.53633	0.7243	0.56611
F1	0.5796	0.65984	0.95464	0.69564	0.59391	0.71258	0.63001
	LAN	ORG	MAI	NMR	ROL	NAME	O
Precision	0.90976	0.81319	0.85327	0.92025	0.77261	0.91135	0.95644
Recall	0.87243	0.70567	0.70835	0.64317	0.74704	0.77482	0.97829
F1	0.88961	0.75484	0.76935	0.74919	0.75792	0.83597	0.96723

TABLE 14: Item level of the dual model: 5-fold evaluation results (Norwegian language).

	ADR	CER	DAT	DEG	DUR	EDU	LOC
Precision	0.76277	0.67949	0.98536	0.60168	0.70851	0.83266	0.81982
Recall	0.63494	0.64647	0.95055	0.50861	0.55324	0.69398	0.66908
F1	0.68934	0.65886	0.96762	0.54826	0.61076	0.75483	0.73366
	LAN	ORG	MAI	NMR	ROL	NAME	O
Precision	0.92364	0.84755	0.96718	0.95344	0.76949	0.92949	0.96403
Recall	0.89205	0.75006	0.76803	0.66007	0.77645	0.91972	0.97934
F1	0.90612	0.79518	0.85348	0.76887	0.7723	0.92289	0.97162

TABLE 15: Item level of the dual model: 5-fold evaluation results (Swedish language).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087913, IEEE Access

	ADR	CER	DAT	DEG	DUR	EDU	LOC
Precision	0.82652	0.80651	0.98318	0.65296	0.80779	0.74805	0.77433
Recall	0.65083	0.63416	0.9401	0.6319	0.66574	0.69927	0.66057
F1	0.71816	0.70327	0.9611	0.61691	0.71359	0.71894	0.70835
	LAN	ORG	MAI	NMR	ROL	NAME	O
Precision	0.94595	0.85436	0.93163	0.95488	0.75279	0.88359	0.95591
Recall	0.90886	0.74626	0.78743	0.78378	0.77253	0.83005	0.97635
F1	0.92611	0.7963	0.85189	0.85722	0.75924	0.85191	0.966

TABLE 16: Item level of the dual model: 5-fold evaluation results (Finnish language).

	ADR	CER	DAT	DEG	DUR	EDU	LOC
Precision	0.88651	0.71375	0.95809	0.46124	0.85787	0.7975	0.87672
Recall	0.78239	0.65574	0.92749	0.55392	0.57111	0.75023	0.71988
F1	0.82916	0.68258	0.94168	0.49325	0.66721	0.77091	0.78792
	LAN	ORG	MAI	NMR	ROL	NAME	O
Precision	0.89563	0.87882	0.96944	0.96927	0.87295	0.91882	0.94663
Recall	0.82253	0.78683	0.85876	0.75938	0.83959	0.92822	0.96949
F1	0.85675	0.82899	0.90768	0.84559	0.85522	0.92188	0.95791

TABLE 17: Item level of the dual model: 5-fold evaluation results (Polish language).

	ADR	CER	DAT	DEG	DUR	EDU	LOC
Precision	0.79919	0.75809	0.97073	0.79097	0.80645	0.78701	0.88899
Recall	0.69683	0.73233	0.94195	0.79031	0.78145	0.78579	0.83883
F1	0.74138	0.74081	0.95607	0.78978	0.79222	0.78466	0.86278
	LAN	ORG	MAI	NMR	ROL	NAME	O
Precision	0.93614	0.8503	0.96603	0.96363	0.76935	0.8741	0.96181
Recall	0.88908	0.75418	0.79916	0.73087	0.82628	0.67643	0.97297
F1	0.91138	0.79892	0.87399	0.82796	0.79642	0.75867	0.96735

TABLE 18: Item level of the dual model: 5-fold evaluation results (English language).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/