A fuzzy logic and ontology-based approach for improving the CV and job offer matching in recruitment process

Article *in* International Journal of Metadata Semantics and Ontologies · January 2021

DOI: 10.1504/IJMSO.2021.120278

2 authors:

Amine Habous
Sidi Mohamed Ben Abdellah University
5 PUBLICATIONS **24** CITATIONS
SEE PROFILE

El Habib Nfaoui
University of Sidi Mohamed Ben Abdellah
146 PUBLICATIONS **1,228** CITATIONS
SEE PROFILE

A fuzzy logic and ontology-based approach for improving the CV and job offer matching in recruitment process

Amine Habous* and El Habib Nfaoui

LISAC Laboratory, Faculty of Sciences Dhar EL Mahraz,
Sidi Mohamed Ben Abdellah University,
Fez, Morocco
Email: amine.habous@usmba.ac.ma
Email: elhabib.nfaoui@usmba.ac.ma
*Corresponding author

Abstract: The recruitment process is a critical activity for every organisation, and it allows to find the appropriate candidate for a job offer and its employer work criteria. The competitive nature of the recruitment environment makes the task of hiring new employees very hard for companies due to the high number of CV (resume) and profiles to process, the personal job interests, the customised requirements and precise skills requested by employees, etc. The time becomes crucial for recruiters' choices; consequently, it might impact the selection process quality. In this paper, we propose a retrieval system for automating the matching process between the candidate CV and the job offer. It is designed based on Natural Language Processing, machine learning and fuzzy logic to handle the matching between the job description and the CV. It also considers the proficiency level for the technology skills. Moreover, it offers an estimation of the overall CV/job offer expertise level. In that way, it overcomes the under-qualification and over-qualification issues in the ICT (Information and Communication Technologies) recruitment process. Experimental results on a ground-truth data of a recruiter company demonstrate that our proposal provides effective results.

Keywords: text mining; natural language processing; feature extraction; metadata weighting; ICT recruitment; fuzzy logic; machine learning.

Reference to this paper should be made as follows: Habous, A. and Nfaoui, E.H. (2021) 'A fuzzy logic and ontology-based approach for improving the CV and job offer matching in recruitment process', *Int. J. Metadata Semantics and Ontologies*, Vol. 15, No. 2, pp.104–120.

Biographical notes: Amine Habous is currently a PhD student in Computer Science at Sidi Mohamed Ben Abdellah University, Fez, Morocco. He received his master degree in Computer Science from Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2016. His research interests focus on the application of text mining techniques to improve the IT recruiting process.

El Habib Nfaoui is currently a Professor of Computer Science at Sidi Mohamed Ben Abdellah University, Fez, Morocco. He received his PhD in Computer Science from Sidi Mohamed Ben Abdellah University, Morocco, and the University of Lyon, France, under a Cotutelle agreement (doctorate in joint-supervision), in 2008, and then his HU Diploma (Accreditation to supervise research) in Computer Science, in 2013, from Sidi Mohamed Ben Abdellah University. His current research interests include Information Retrieval, Language Representation Learning, Machine learning and Deep learning, Web mining and Text mining, Semantic Web, Web services, and social networks. He has published in international reputed journals, books, and conferences, and has co-edited eight conference proceedings and special issue books. He has served as a reviewer for scientific journals and as program committee of several conferences. He is a co-founder and Chair of the IEEE Morocco Section Computational Intelligence Society Chapter.

1 Introduction

The swift development of modern technologies of Information and Communication leads to an increase in the number of companies who use web portals to hire their employees, since placing job advertising on websites is a lot cheaper than other media such as newspapers or magazines (Cabrera-Diego et al., 2019). Thus, the number of candidates for a given job becomes challenging to handle by the company recruiters (Catherine et al., 2010). Besides, the traditional matching that involves treating each candidate's CV, on the one hand, does not work anymore. Matching a candidate to a job post could be sometimes crucial according to its nature. Consequently, the recruiter gets stressed, and the selected candidates might be not the suitable ones. The automation of this matching has been the subject of many works in text mining and other sciences of data fields (Janusz et al., 2018; Converse et al., 2004) in order to resolve this problem and deal with other tasks such as scoring and ranking job applicants. The mismatching occurs if we don't treat the under-skilling and the over (under) qualification issue. The under-skilling means the situation in which an individual lacks, in a particular way, the required skills and abilities to perform on the current job. The over (under) qualification is the situation in which an individual has a higher (lower) expertise level than the current job requires, according to the candidate overall experience and realised job tasks. The mismatching impacts candidate productivity. Therefore, it leads to real economic and social losses.

Text mining is the process of analysing the unstructured text data to extract the meaningful information from the hidden patterns. It gathers all the process phases, starting with textual document retrieval and natural language processing to text classification and clustering. The free textual nature of job description and CV treated by the system makes the application of text mining tools very efficient.

The Natural Language Processing (NLP) is a set of techniques to make the computer able to understand texts using human knowledge and advanced computing. Khurana et al. (2017) classified NLP into two parts i.e., Natural Language Understanding and Natural Language Generation, which evolves the task to understand and generate the text:

Linguistics is the science of language that includes Phonology that refers to sound, Morphology word formation, Syntax sentence structure, Semantics syntax and Pragmatics which refers to understanding. The technologies of natural language processing are used to ensure many tasks in our retrieval system building. Such as tokenisation, stemming, lemmatisation and POS tagging.

The contributions of this work are summarised as follows:

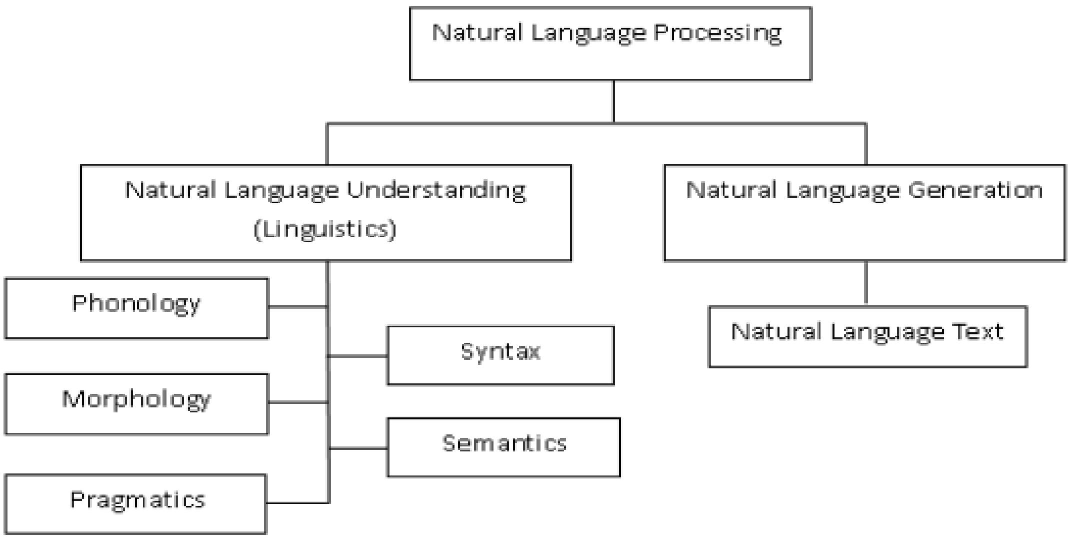
- We propose an information retrieval system to automate the matching task.
- We create and weight metadata using fuzzy logic inference.
- To further enhance the overall performance, we propose an ontology-oriented ICT for inferring new features from CV/Job offer representation and then perform classification according to expertise level.
- We propose a matching algorithm for scoring and ranking the results.
- We conduct extensive experiments on a ground-truth data of a recruiter company. Experimental results demonstrate that our proposal provides effective results.

The rest of this paper is organised as follows. Section 2 presents the related work. Section 3 presents the preliminaries. Section 4 describes our proposal. Section 5 details and discusses the experiments. Finally, Section 6 concludes this paper.

2 Related work

Several e-recruitment systems have been proposed to facilitate the recruitment process, by automating the matching task between the job seekers' CVs and the recruiters' offers. In previous techniques, domain ontologies have been commonly used, such as ER-Ontology (Yahiaoui et al., 2006) which is inspired from common parts between job offers and CV for semantic annotation. Therefore, (Bourse et al., 2002) used three types of ontologies for CV and job offer annotation: sectors, skills and enterprises. Desmontils et al. (2002) used linguistic ontology for indexation and structuring the candidate CV. Besides, some works used machine learning techniques for computing similarity between CV and job offers. For example, using learning to rank algorithms (Van Belle and Foster, 2018). Faliagka et al. (2002) used the same algorithm after extracting candidate personality treats based on their blog. In many works, the CV always represents the element of answer for a job offer. However, the proposed system Automated Job Search (AJS) in Al-Ramadin et al. (2015) has a reciprocal income. It consists of algorithms that use a database of job classification and job specifications to match between the input CV and the job offers.

Figure 1 Classification of NLP


Natural Language Processing
Natural Language Understanding (Linguistics)
Natural Language Generation
Phonology
Morphology
Pragmatics
Syntax
Semantics
Natural Language Text

Fuzzy logic has been used widely due to its ability to solve a wide range of problems successfully. Many contributions to the human resources field proposed fuzzy systems to enhance the selection process. Balas-Timar and Ignat (2015) proposed a fuzzy expert system for candidate selection taken into account three criteria: education, previous professional experience and motivation. Klosowski et al. (2016)designed a fuzzy controller to aid the decision-making process using the Mamdani-type fuzzy inference. The main idea of this contribution is to convert the qualitative features into quantitative ones to compare workers regarding the production tasks. Samaila et al. (2018) developed a fuzzy framework for the personnel selection process using the Fuzzy Simple Additive Weighted (FSAW) method. This method is also known as weighted linear combination or scoring methods, which is a simple and most often used multi-attribute decision technique. An amalgamated method of recruitment was proposed in Khandelwal and Agrawal (2015), by the use of fuzzy logic and genetic algorithm with the Hungarian method.

In this paper, we present a retrieval system for automating the matching process between job offer and CV in the ICT recruitment domain. The main and specific feature of our work is interpreting the recruiter need, and the candidate profile treats by the extraction of the technical competencies, with their proficiency levels. Our proposed system uses a fuzzy inference mechanism to infer new technology skills to handle the lack of information that could occur in the job description. This method enhances the matching score, at the same time, avoids the mismatching caused by the under skilling. In parallel, we propose a CV/Job offer classification according to their overall expertise level to prevent the mismatching that could occur in the case of under (over qualification).

3 Preliminaries

This section presents the necessary background for understanding the remainder of this paper, and it includes:

- Problem definition
- Ontology
- Fuzzy logic
- Tokenisation and sentence splitting
- Named entity recognition
- POS tagging

3.1 Problem definition

In this paper, we dealt with Information Retrieval (IR) of text documents in the recruitment context. Let $C = \{c_1, c_2, ..., c_m\}$ and $J = \{j_1, j_2, ..., j_n\}$ be the sets of CV and job offers, respectively. This paper aims to define the similarity function S: $S(c, j) : J \times C \rightarrow [0,1]$ to retrieve the CV and rank them according to a given job offer.

3.2 Ontology

The idea behind the semantic web is to bring a standard structure to web pages, providing their content with meaning which allows the external applications to carry out sophisticated tasks (Kuck, 2004) such as information retrieval. The ontology plays the primary role in making machines capable of understanding the semantic of languages that humans use (Fensel et al., 2000). Ontology is defined as an explicit formal specification of a shared conceptualisation. It provides semantic meaning through relations between concepts (Zhihong et al., 2002). The Resource Description Framework (RDF) is introduced as an underlying framework to use ontology. The RDF data model treats each piece of information as a triple: subject-property-object (Lassila and Swick, 1999). In Computer science, ontology designates different objects depending on the context (Roussey et al., 2011). For example:

- Thesaurus in the field of information retrieval
- Model represented in OWL in the field of linked-data
- XML schema in the context of databases

The distinguishment between the types of the ontology helps to clarify their content, their use and their goal.

3.3 Fuzzy logic

Fuzzy logic is a mathematical logic model in which the truth is partial. It is based on approximating reasoning instead of exact reasoning (Goswami and Shishodia, 2013). This model consists of the fuzzy set theory proposed by Zadeh (1965), as an extension of the classical definition of a set (Dubois and Prade, 2005). In fuzzy logic, the membership function that maps each point in the input space to a membership value is between 0 and 1; contrarily, in classical logic equals 1 or 0. Thus, in fuzzy logic, the set boundaries become no more clear (Klement and Mesiar, 2005). Fuzzy inference systems were developed to control complex processes using the human experience, from a set of linguistic rules, in which words are modelled by fuzzy sets (Zwick, 1993). Generally, three types of fuzzy inference methods are proposed in the literature: Mamdani fuzzy inference, Sugeno fuzzy inference, and Tsukamoto fuzzy inference (Wang, 2015).

3.4 Tokenisation and sentence splitting

Tokenisation plays a significant role in Natural Language Processing (NLP), it aims to separate a piece of text into smaller units called tokens. The tokens could be words, numbers or punctuation marks which are created by specifying the words boundaries. We use these boundaries to mark the ending point of a word and the beginning of the next word. These tokens are considered as a first step for stemming and lemmatisation. Sentence splitting consists of assembling the tokens to form sentences. It also requires defining the beginning of these sentences and their endpoint where punctuation marks are often used. We used this technique for the job task extraction which is an essential task in our matching system.

3.5 *Named entity recognition*

Named Entity Recognition is a task in Information Extraction (IE) that aims to identify specified entities in a body or bodies of texts. Since we work in the ICT recruitment field, we used a domain ontology to identify the named entities in free text.

3.6 *Part of speech (POS) tagging*

The POS method consists of a grammatical analysis to label the words according to their nature and the syntactic relations between them (Toutanova and Johnson, 2012). It is beneficial in NLP; it accomplishes so many tasks as building lemmatisers which are used to reduce a word to its root form. The POS tagging step is essential to understand any sentence meaning or extract relationships and make knowledge graphs. We could list many types of tagging techniques:

- *Lexical-based methods*: Assigns the POS tag the most frequently occurring with a word in the training corpus (Voutilainen, 2003).
- *Rule-based methods*: Assigns POS tags based on rules. For example, we can have a rule that says, words ending with "ed" or "ing" must be assigned to a verb. Rule-Based Techniques can be used along with Lexical-Based approaches to allow POS Tagging of words that are not present in the training corpus but are there in the testing data (Brill, 1992).
- *Probabilistic methods*: This method assigns the POS tags based on the probability of a particular tag sequence occurring. Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) are probabilistic approaches to assign a POS Tag (Gimpel et al. 2010).
- *Deep-learning methods*: Recurrent Neural Networks can also be used for POS tagging (Santos and Zadrozny, 2014).

4 Methodology

Figure 2 shows the overall proposed architecture. In the sub-sections below, we explain the module in detail. First, we present the domain ontology used in our IR system and we explain the proposed method that we use to weight the relationships in this ontology. In the next sub-section, we describe the text pre-processing, features extraction and inference tasks. The third sub-section explains the matching algorithm that we used to compute the similarity between job offers and CV. In the last sub-section, we present the method that we used to build the classification model to classify the job offers and CV according to the expertise level.

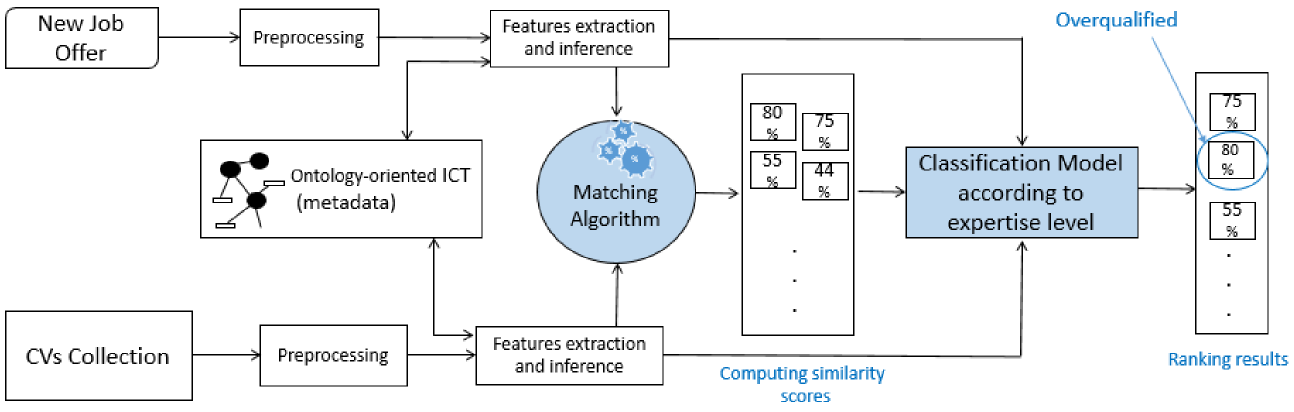
4.1 *The proposed ontology-oriented ICT competencies and concepts*

The use of a domain ontology in our retrieval system is significant. It ensures two principal functionalities:

- First, gathering the recruitment experts knowledge: make the matching system able to detect technology skills and the other ICT important concepts.
- Second, the system becomes able to handle the synonymy and hyponymy relations between the concepts. Moreover, we use the relationship weight between the technology domain and skill for CV and job offers representation; in the end of this section, we explain in detail the method that we use for weighting those relationships. This ontology ensures an important part of intelligence in text processing.

In the first phase of the system construction, we were working on building an ontology that would cover all the essential knowledge in the ICT field. We used knowledge bases that already exist in the ICT field such as O * net to extract the technology skills and technology domains. Moreover, we used the data provided by our partner which is a company specialising in recruitment.

Figure 2 The overall matching system


New Job Offer
Preprocessing
Features extraction and inference
Overqualified
Ontology-oriented ICT (metadata)
Matching Algorithm
80 %
75 %
55 %
44 %
Classification Model according to expertise level
75 %
80 %
55 %
CVs Collection
Preprocessing
Features extraction and inference
Computing similarity scores
Ranking results

4.1.1 The ontology components

Classes:

- *Technology skill*: this class presents the set of technology skills in the ICT domain, its instances are updated according to the ICT recruitment dynamicity.
- *Domain*: predefined set of technology skills that coexist to be used in the same job. Example: "Front end" (technology skills: “Css”, “React” and “Ajax” ...)
- *Sector*: the activity sector is used to filter the results
- *Position*: Job position that could used also for filtering the results.
- *Concept*: technical keywords could be global or linked to specified domains.

Relationships:

- *have_technical_skill*: it binds a technology skill with a specific domain (ex: “big data” have_technical_skill “Hadoop”)
- *have_domain*: the reciprocal relationship of have_ technical_skill
- *have_pos*: it binds a given job position to a domain (ex: “web” have_pos “full stack developer”)
- *include*: (reflexive) domain include its sub-domain (“web” include “front-end development”)
- *have_concept*: binds a technical term to its domain
- have_synonym links that binds the classes’ instances with text are used to represent the synonymy relationship.

Figure 3 presents the proposed ontology-oriented ICT competencies and concepts structure:

4.1.2 Weighting relationship between technology skills and domains using fuzzy logic

In this section, we will present the method for weighting the relationship between technology skills and domains. This task aims to ensure the inference of the key technology skills of a given domain. Thus we could use them to enhance the matching between the technology skills cited in the job offer and those presented in the CV. Especially when the recruiter does not explicitly specify them.

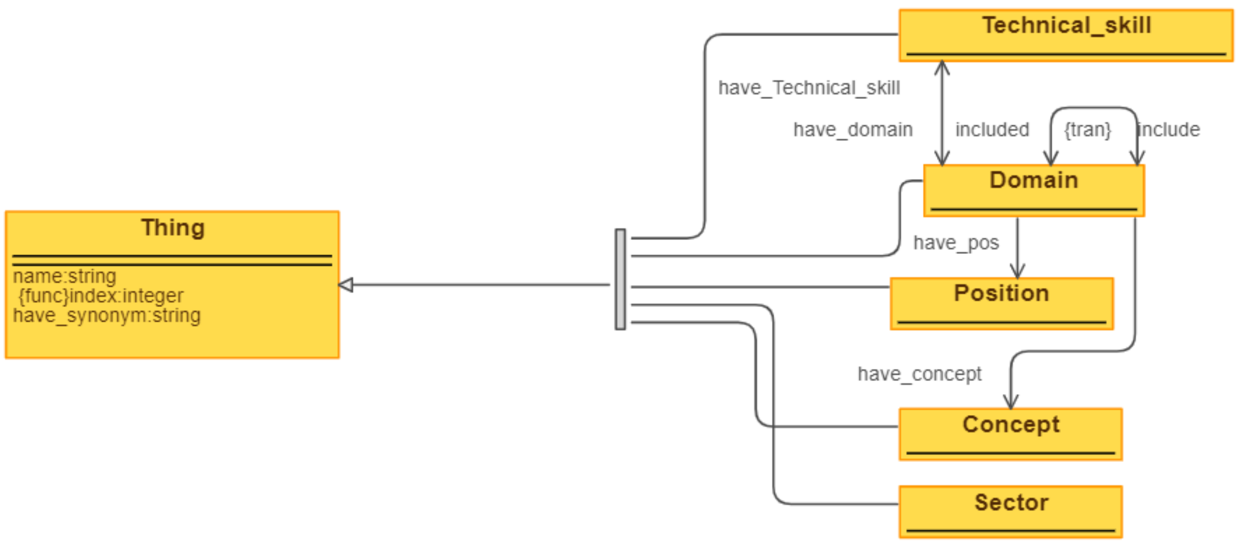
The idea behind proposing a fuzzy logic-based model to weight the relationship between technology skills and domains is that we need to deal with numerical data and linguistic knowledge. In what follows, we describe the two parameters that the fuzzy control takes as inputs to weight the relationship between the technology skill an domain in the ontology.

- *Input 1*: To determine to what extent a technology skill relates to a technology domain we calculate its document frequency in job offers in which that domain occurs according to the technology domain. If this frequency is high, we deduce that there is a strong correlation between them. For example, the technology skill: ‘Python’ for the domain: ‘Machine learning’ and ‘CSS’ for ‘Front end’. We use equation (1) to calculate the document frequency of a technology skill ‘*s*’ according to a technology domain ‘*d*’.

$$f_d(d,s)=\frac{n_{d,s}}{n_d} \tag{1}$$

where ‘ $n_{s,d}$ ’ is the size of the job offers set containing both the technology skill ‘ s ’ and the domain d , and n_d the size of all the job offers containing the technology domain ‘*d*’.

Figure 3 Ontology-oriented ICT competencies and concepts



- *Input 2*: the technology skill document frequency in job offers in which that domain occurs according to the technology skill. We use equation (2) to calculate the second parameter.

$$f_s(d,s) = \frac{n_{d,s}}{n_s} \tag{2}$$

n_s the size of all the job offers containing the technology skill '*s*'

Let's take an example to show the importance of using these two parameters:

$\{J_1, J_2, J_3\}$ the job offer sets containing the domain 'Machine learning', 'Python' and 'English language' respectively. Let's consider:

- $\Omega(J_1) = 200$: job offers that contain the technology domain 'Machine learning'
- $\Omega(J_2) = 250$: job offers that contain the technology skill 'Python'
- $\Omega(J_3) = 900$: job offers that contain the technology skill 'English'

and $\Omega(J_1 \cap J_2) = 150$, $\Omega(J_1 \cap J_3) = 190$

where Ω is the set cardinal number.

now let's compute the frequencies according to J_1:

$f_d(m.learning)(Python) = 0.75$

$f_d(m.learning)(English) = 0.95$

If we only rely on the first parameter, we deduce that 'Python' correlation is lower than the 'English' one according to 'Machine learning' because the document frequency of 'English' is high. We can easily handle this issue by adding the second parameter:

$f_s(m.learning)(Python) = 0.6$

$f_s(m.learning)(English) = 0.2$

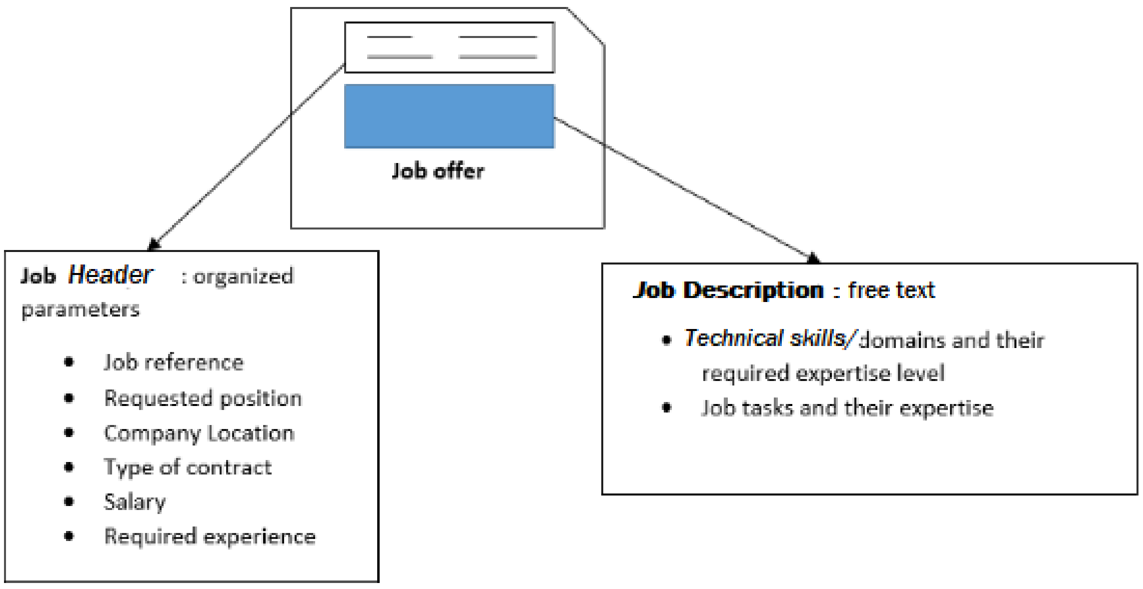
We will use those parameters as inputs for the fuzzy inference to deduce the final weight. We use a mechanism to update that weight according to the new job offers treated by the matching system.

4.2 *Text features extraction and inference from CV and job offer*

4.2.1 *Job offer structure*

In recruitment announcements, the job offer is a half-structured textual document. It is composed of two parts; the first part is the job header which is a set of information about the position, such as the type of contract, the salary,… and information about the company. The second part is the job description containing a free text that describes the employer needs. We can schematise this description in Figure 4.

Figure 4 Job offer structure


Job offer
Job Header : organized parameters
Job reference
Requested position
Company Location
Type of contract
Salary
Required experience
Job Description : free text
Technical skills/domains and their required expertise level
Job tasks and their expertise

4.2.2 Text pre-processing

We used the traditional document pre-processing in NLP (Kannan and Gurusamy, 2014):

- Tokenisation: dividing the job offer (CV resp.) to tokens
- Filtering: removing terms that are meaningless in any language
- Stemming: the transformation of any term to its stem to unify the document units, for example, process, processing and processed will be stemmed to "process".

4.2.3 Job offer and CV textual features

In this step, we aim to transform the job offers and CV into a numerical representation. There are three types of features that we extract from job offers and CV:

- *Technology skills*: We extract the list of the technology skills along with their weights that present their proficiency level. We calculate these weights differently from CV and job offers; we will explain how we compute them in the next subsections.
- *Technology domains*: The list of the technology domains with their weights
- *Job tasks*: In recruitment, job tasks are essential to know whether a candidate corresponds to a job post or not. Therefore, we took into consideration this feature in our matching system. Technically, we divide the recruitment documents into phrase tokens; and afterward, we identify the job task using the activity verbs that we will explain below.

We separate the numerical representation of job offers and CV to compute the similarity into three levels. Thus we can have the overall score of similarity and the partial value of similarity for each part. Moreover, we can set up the general computation by assigning coefficients to each level.

4.2.4 Technology skills extraction

a) *Extraction of technology skills and their required proficiency level (weight) from job offer:* After obtaining the pre-processed terms, we use the ICT ontology to identify the technology skills from free text. Then, we extract each skill weight that represents its importance. In any job offer, the employer has two ways to express the priority of a technology skill or the proficiency level that a candidate should have. Either by:

- Required years of experience
- Expression of proficiency level

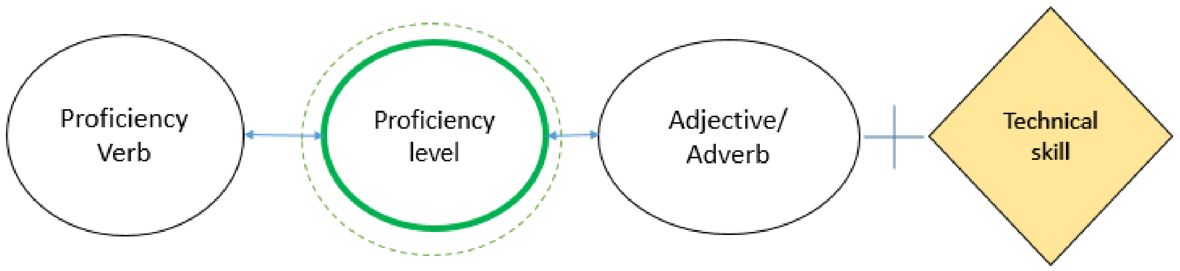
Required years of experience: The employer could directly specify the number of the required years of experience. It is the easiest way to compute the technology skill weight in the job offer. Technically, we identify the pattern containing the term that represents 'years' after a given number in the same token phrase. For example, you must have at least 5 years of experience using Adobe XD

Expression of proficiency level: If the employer does not specify the number of experience years, we analyse linguistically the phrase token containing the technology skill to extract the weight. This weight is always positive or neutral (the recruiter does not specify his need). Technically, the value assigned to this weight is computed according to text units contained in the same sentence. In what follows, we explain the process of weighting the technology skills.

Figure 5 shows the text units that we analyse for weighting the technology skill. Each one has its value that identifies a degree for the recruiter need. For example, 'to use PHP' and 'to master PHP' are very different. The second expression means that the employer needs someone that uses 'PHP' very well. Consequently, the weight of 'PHP' in this expression is high than the first one. Sometimes the level of proficiency is directly mentioned; in this case, we ignore the proficiency verb. The adjective or adverb is used to enhance the weight. For example:

1. 'must have a good mastery...' we mentioned the level of proficiency directly.
2. 'to know well...' know is the proficiency verb. Moreover, 'great' and 'well' are adjectives that enhance the skill importance in the job offer; consequently, we improve the weight. These units are extracted automatically from the job offer test corpus using the POS tagging method. We have created a dictionary in which we assign a weight to each linguistic unit used for the final weight computation. This assignment is based on the semantics of these units as we saw above. Algorithm 1 summarises the steps to compute the technology skill weight for the two cases.

Figure 5 Expression of proficiency



Algorithm 1: The computation of the technology skill weight

```
1: Extracting the technology skill using the ontology
2: Identify the phrase token using punctuation marks
3: $Weight_{Skill} \leftarrow 1$                          ▷ Initialisation
4: if required years of experience is given = True then
5:     $Y \leftarrow$ number of years
6:     if $Y \geq 10$ then $Weight_{Skill} \leftarrow 7$
7:     else if $10 > Y \geq 5$ $Weight_{Skill} \leftarrow 5$
8:     else if $4 \geq Y \geq 3$ $Weight_{Skill} \leftarrow 3$
9: else
10:    if proficiency level mentioned = True then
11:        if Adjective/Adverb mentioned = True then
12:            $Weight_{Skill} \leftarrow Weight_{PL} + Weight_{Ad}$
13:    else
14:        $Weight_{Skill} \leftarrow Weight_{PL}$
15:
16: $Weight_{PL}$ : is the assigned weight to the term that represents the proficiency level (use, mastery...)
17: $Weight_{Ad}$ : the assigned weight to adjective / adverb.
```

Technology skills inference: We use the relation weights that link the technology skills and the domain of the ICT ontology. To infer the required skills that the recruiter did not mention. We explained in detail the method we used to weight these relationships in the ontology section above.

Table 1 summarises the weight ranges used for technology skill proficiency levels according to the extraction method.

Table 1 Ranges of the proficiency levels weight

Weight ranges	*Technology skill extraction method*
$\geq$ weight ≥ 3	Required years of experience
$>$ weight ≥ 1	Expression of proficiency level
$\geq$ weight ≥ 0.5	Technology skills inference

- The first method describes in a certain way the proficiency level of the skill, which is why we assigned the higher weight which gives priority to the technology skill over the job offer.
- We used the linguistic expression that the recruiter provides to express the level of expertise necessary for the skill. Therefore, this method is based on a semantic interpretation of the recruiter expression. Thus we have parameterised the unit weights of the linguistic units so that the resulting weight has less impact than the weights assigned using the first method.
- We used the last method to infer the technology skill that the recruiter did not mention, yet we find in the job offer a domain that requires the mastery of this skill. Thus, we decided that the weight of the relation between the skill and domain in the ICT ontology must be weak compared to the other extraction methods.

b) *Extraction of technology skills and their required proficiency level (weight) from CV:* The CV representation has almost the same characteristics as the job offer regarding technology skills. However, the weight calculation is different; given the nature of the CV, the candidate can represent his technology skill proficiency level through the technical environments of the missions he has carried out. Therefore, we could use the occurrence number of the skills in the CV for the weight assignment.

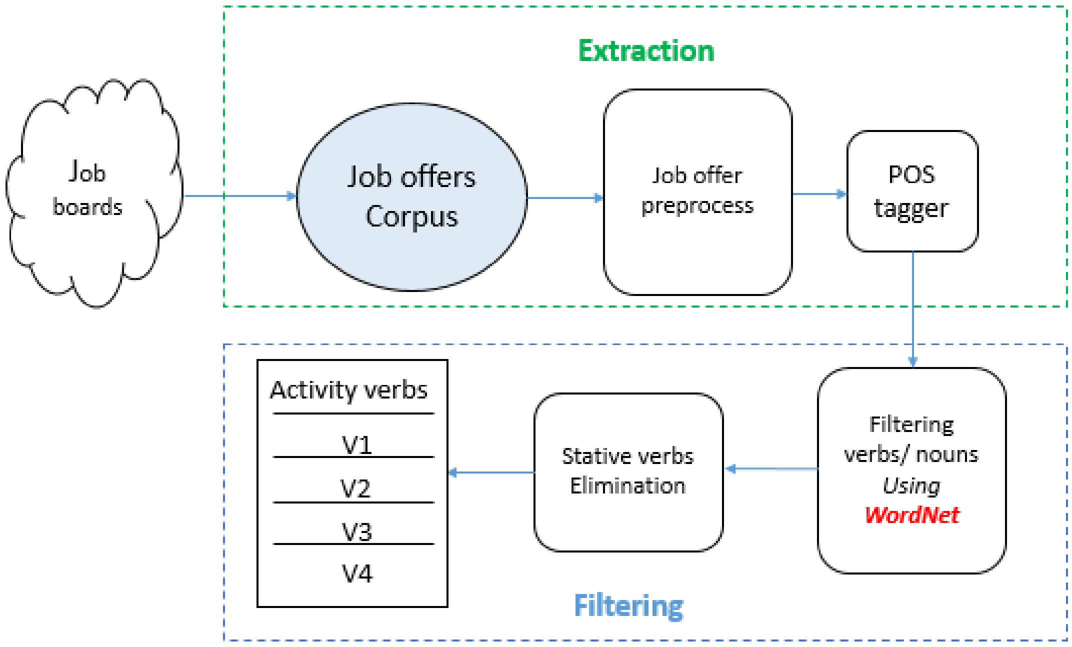
4.2.5 Technology domains extraction

The recruiter mentions the technology domains in job offers and resumes in the same way as technology skills. Thus, we use the same method explained above for their representation. The particularity of their extraction is that we take into consideration the notion of semantics using the ontology. For example, if the recruiter requires a level of expertise for the domain 'BI', it is matched with the term 'Business intelligence' or 'Data analysis' in the candidate's CV.

4.2.6 Job tasks extraction

In a job offer, the employer quotes the job tasks which the candidate should accomplish. Thus, the job task extraction is beneficial to the matching process for computing the similarity. To extract the candidate job tasks from its CV or the required ones in job offers, we should first create a database to store all the activity verbs in these elements. We use the activity verbs to identify a specific job task. For example: 'Managing the project work' where 'to manage' is the activity verb for this job task. Here, is the proposed schema to extract activity verbs from the job offer corpus (see Figure 6):

Figure 6 The activity verbs base creation process



- *Extraction step*: We use the Part of Speech (POS) tagging method for identifying the verbs form from the job offers. Since the objective is recognising the verb form, we chose a rule-based POS tagger, is available and easy to apply.
- *Filtering step*: One of the rule-based tagging method limits is interpreting some technical terms as verbs due to the word morphology. For example, the term 'Docker,' which refers to technical skill in our context, is identified as a verb if we use a French POS tagger. To overcome this limit, we use the lexical tool WordNet, which is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, and adjectives are organised into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets (Miller et al., 1993). Basing on this lexical base, we filter the words having a verb morphology identified by the POS tagger and an empty synset (synonym set). In other words, we eliminate the terms that have no sense in the language that we use. After that, we use a predefined set of stative verbs to filter them.

After creating the activity verbs list, we use them to identify the job tasks sentence tokens. Then, we add them to a list as a feature for CV and candidate representation.

4.3 *The matching algorithm*

The matching algorithm computes the similarity between CV and job offer on three levels according to the features that we extract from the text. Each part has a partial value, which we use to deduce the overall similarity score.

4.3.1 *The similarity between CV and job offer: technology skills*

We compute the technology skills partial similarity between the CV and the job offer using Algorithm 2. We take technology skills that we extract from both job offer and CV as inputs.

Algorithm 2: How to compute the technology skills similarity between job offer and CV

```
1: inputs: $skills_{offer}$, $skills_{cv}$        ▷ lists of extracted
   skills from CV and job offer
2: $score_{skill} \leftarrow 0$
3: for $skills_o \in skill_{offer}$ do
4:     if $skills_o \in skills_{cv}$ then
5:         $score_{skill} \leftarrow score_{skill} + \min\left(1, \frac{w_{cv}}{w_o}\right)$   ▷ $w_o$ and
           $w_{cv}$ the weights that correspond to $skill_o$
6: $score_{skill} \leftarrow \frac{score_{skill}}{Size\left(skills_{offer}\right)}$
```

4.3.2 *The similarity between CV and job offer: technology domains*

We calculate the partial similarity of technology fields using the previous method. As inputs, we take the lists of the technology domains that we extract from both job offer and CV. As we said before, the synonyms of a given domain are represented by its unique concept in the ontology.

Algorithm 3: The computation of the technology domains similarity between job offer and CV

```
1: inputs: $domains_{offer}$, $domains_{cv}$    ▷ list of extracted domains of both CV and job offer
2: $score_{domain} \leftarrow 0$
3: for $domain_o \in domains_{offer}$ do
4:     if $domain_o \in domains_{cv}$ then
5:         $score_{domain} \leftarrow score_{domain} + \left(1, \frac{w_{cv}}{w_o}\right))$
           ▷ $w_o$ and $w_{cv}$ the weights that correspond to $domain_o$ and the same domain from the CV (respect.)
6: $score_{domain} \leftarrow \frac{score_{domain}}{Size\left(domains_{offer}\right)}$
```

4.3.3 *The similarity between CV and job offer: job tasks*

To compute the similarity between the job offer tasks and those of CV, we look for the CV tasks that correspond to one of the job offer tasks. Therefore, we rely on the comparison of the terms that compose the job tasks. We use Algorithm 4 to compute the job tasks similarity between the job offer and CV. The calculation of this similarity is also taking into consideration the semantic relationships from the ontology. Thus, the terms of each class concept, job positions, and sectors are unified in the extraction step.

Algorithm 4: The computation of the job tasks similarity between a job offer and CV

```
1: inputs: $tasks_{offer}$, $tasks_{cv}$    ▷ list of extracted tasks of both CV and job offer
2: $score_{task} \leftarrow 0$
3: for $task_o \in tasks_{offer}$ do
4:     $Scores_{task} \leftarrow [\,]$
5:     for $task_{cv} \in tasks_{cv}$ do
6:         Compute Jaccard Similarity between terms in
7:         $task_{cv}$ and $task_o$
8:         Append the result to $Scores_{task}$
9:     $score_{task} \leftarrow score_{task} + \max\left(Score_{task}\right)$
10: $score_{task} \leftarrow \frac{score_{task}}{Size\left(tasks_{offer}\right)}$
```

We used Jacquard coefficient in the similarity calculation to compare between the terms from the two sets to see which members are shared and which are distinct. It is a measure of similarity that ranges from 0 to 100%. The higher the percentage, the more similar the two sets.

4.4 *Proposed method for Job offer/CV classification according to their expertise level*

While a candidate may have all the requirements set by a recruiter, it is possible that the latter asks for a candidate with a certain level of experience and the system returns results that exceed or vice-versa the level requested. Our method aims to identify the level of expertise of an offer or a CV by analysing the activity verbs that we extract to identify the job tasks. For example, it's clear that the job task: 'managing the project work' shows a high expertise level, according to the activity verb 'to manage'. The idea is to extract the correlation between these activity verbs and the minimal experience required in the job offer (the candidate experience regarding to CV), that afford us a great idea about the global expertise level. Since the candidate experience extraction from CV is a very complicated task for the machine.

In this section, we will present the proposed solution to classify the job offers and CV using the activity verbs as features.

4.4.1 *The classification model architecture*

Figure 7 describes in depth the proposed model for job offers and CV classification according to the expertise level. This model provides us a great tool to avoid the over-qualification and under-qualification issues. The proposed model is composed by three sub-models, the first one is used for activity verbs classification according to the required experience in the training set of job offers. The second one, is the model that we build for the job offer/CV classification using the activity verbs as features; in order to create a corpus of the job offers that are more likely to require a high expertise level. The last model is based on a fuzzy inference for the final classification of the job offers. The sub-sections above describe each model in details.

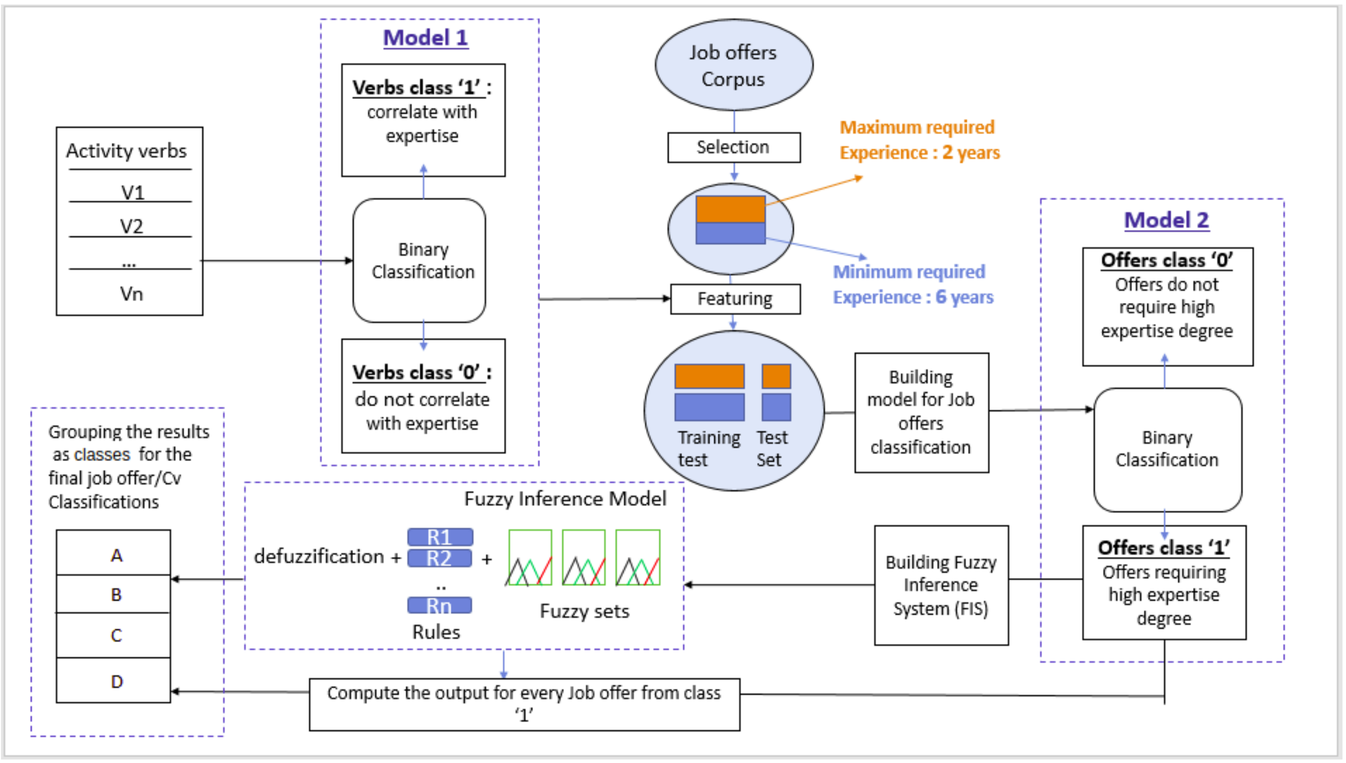
4.4.2 *Activity verbs classification*

The first step in building the general model, is to find the correlation between the activity verbs and the experience required in each job offer in the data set. Thus we could classify them to two classes:

- class '0': verbs do not correlate with the expertise level (for example: to develop, to participate..)
- class '1': high expertise activity verbs (for example : to manage, to pilot)

In the learning phase, we represent each activity verb with its document frequency according to the required experience of the job offers in which occurs. Thus we choose a spatial representative vector V_{verb} for each activity verb that represents this values

Figure 7 The classification model architecture


Activity verbs
V1
V2
...
Vn
Model 1
Verbs class '1' : correlate with expertise
Binary Classification
Verbs class '0' : do not correlate with expertise
Job offers Corpus
Selection
Maximum required Experience : 2 years
Minimum required Experience : 6 years
Featuring
Training test
Test Set
Building model for Job offers classification
Model 2
Offers class '0' Offers do not require high expertise degree
Binary Classification
Offers class '1' Offers requiring high expertise degree
Grouping the results as classes for the final job offer/Cv Classifications
A
B
C
D
Fuzzy Inference Model
defuzzification +
R1
R2
..
Rn
Rules
Fuzzy sets
Building Fuzzy Inference System (FIS)
Compute the output for every Job offer from class '1'

We calculate the verb document frequency is obtained as follows:

$$df_{(Y)}(v) = \frac{n_{vy}}{n} \tag{3}$$

where n_{vy} is the number of job offers that require the experience level Y and contain the activity verb 'v'. n is the size of the corpus.

Thus, activity verb vector is written as:

$$V_{verb}\left[df_{(1-2)}(v), df_{(3-4)}(v), df_{(5-10)}(v), df_{(10+)}(v)\right]$$

We chose the four levels of experience (Y) based on the response of an expert in the field of IT recruitment.

We identify the training set from those verbs using their semantics for the very apparent ones. Then, we annotate them with (0 or 1). We used a logistic regression for their classification. Finally, we store them with their weighting sum values according to the results of the classification model to be used as features for the next step.

4.4.3 Classification of the job offer according to the level of expertise using the required experience

This second step aims to classify the offers to two classes using the activity verbs as features:

- *High expertise level class '1'*: offers are more likely to request high expertise. In the training phase, we use the job offers from the data set, requiring an experience that exceeds six years.
- *Beginner class '0'*: offers that do not require experimented candidates. We used job offers from data set where the minimum experience required is lower than two years.

The features used in this classifications are:

- *Mean of weighting sum*: The first feature is the mean of the weighting sum of the first-class verbs occurring in the job offer.
- *Proportion*: We divide the number of first-class verbs by all the verbs extracted from the job offer.
- *Deviation*: we calculate the standard deviation of first-class verbs in the job offer.
- *Profile degree*: We use this feature to enhance the classification precision. For example, the job recruiter could not write any correlated verb for a job offer requesting a manager. The manager degree can handle this exception. Consequently, the precision is enhanced; especially, we work in a noisy environment.

The model built helps us to show the efficiency of our proposed solution. We use it in job offer classification according to activity verbs in the last step.

4.4.4 Building the fuzzy inference system (FIS)

In this step, we built a fuzzy inference engine. It takes three parameters as input:

- the mean of weighting sum
- Proportion
- Deviation

Those inputs are the same as the features of the previous step (except the profile degree). We study the classifier results to create these parameters fuzzy sets. The fuzzy engine output is the expertise level weight of the job offer according to the activity verbs. The last step is the fuzzy inference application for the job offers from the first class (High expertise job offers) and clustering them into four ordered groups: 'A'–'B'–'C'–'D' according to the resulted weights values.

4.4.5 New job offers/CV classification according to the expertise level

We will present the steps for classifying new job offers/CV using the model above.

Step 1: In the features extraction step, we obtain the list of the activity verbs (Sub-section 4.2.5). Thus, we use the first class activity verbs to calculate the parameters for classifying the job offers/CV.

Step 2: Use those parameters as inputs for model 1 to classify the job offer:

- If the resulted class is '0' we assign directly the last grade '*E*' to the current Job offer/CV.
- if the resulted class is '1' classification steps continue

Step 3: we take the parameters as inputs to the fuzzy inference model to compute the output of the model.

Step 4: we compare the output value with the pre-computed ones of the job offers in the training set. Finally, we assign the right class 'level of expertise: A – B – C – D' to the job offer/CV.

5 Results and discussions

In this work, we used the Python language for implementing and evaluating our algorithms on a HP 4540s computer with an Intel i5 CPU running at 3.20 GHz with Ubuntu 18.04 (64-bit) and 8 GB of RAM. We used the following libraries for implementing the main tasks:

- *NLTK*: NLP library
- *Owlready2*: Ontology-oriented programming
- *Apache Tika*: Text extraction from PDF / DOCX documents
- *Scikit-learn*: Machine learning
- *SKfuzzy*: Fuzzy logic implementations

5.1 Corpus

The first step for building the matching system requires data capturing from job boards to create our corpus. We fill it up with job offers because it is free to obtain them. However, the job boards do not publish their CV bases. Consequently it is challenging to create a CV corpus. We worked in French employment domain. Thus we captured about 10,000 document from E-recruitment sites that share the job offers in France.

5.2 The results of weighting the relationship between the technology skills and domains

Table 2 gives examples of the technology skills resulted weights according to domains to show the efficiency of our proposed method. This table contains just a portion of the technology skills instances, not all the resulted ones. We also mention that the skills that have the weight 0.5 or less are not inferred as key skills according to a given domain.

Table 2 Examples of the resulted weights between technology skills and domains

Technical competence	*Domain*	*Weight*
Agile method	Full stack	2
Python	Machine learning	2
Python	Big data	1.5
Sql	Data base	1.49
Rest services	Front End	1.35
R	Statistics	1.35
Css	Front end	1.33
MongoDb	Nosql databases	1.17
Php	backend	1.07
C	embedded systems	1.05
Docker	Devops	1.03
React	Front end	1.01
Linux	Systems administration	0.75
Angular js	Front end	0.55

5.3 The matching algorithm results

In this section, we present the results of the proposed matching system. The results as we said below are obtained in French recruitment field.

5.3.1 Data set

Since we could not find a standard data set in recruitment to evaluate the performance of our system, the recruiter company proposed a data set combining some relevant CV and job offers from the ICT recruitment field in French. In the following, we detailed the two sets that we used in the evaluation step:

Table 3 describes the job offers data set that we used for evaluating the system performance. Since the system is a content-based solution, we used the job offer required profiles to define the job offers fields. We compare the retrieved CV occupied job posts with the requested job offers domains to evaluate the system performance.

Table 3 The Job offers used for the tests

Job position field	*Data set size*
Data science	68
Big data development	45
Business intelligence	76
Front-end development	41
Back-end development	27
Java / JEE development	9
Total	266

Table 4 presents the CV set used to test the performance of our matching system. It contains detailed information for each CV. The last column lists the job positions that the candidate occupied during his work experience.

5.3.2 Evaluating the matching system

Table 5 describes the matching system results, calculating the average precision and recall for each job offer field (requests). We use precision and recall measures to show the effectiveness of our proposed system. The graph in Figure 8 describes the variation in precision and recall for each job offers field.

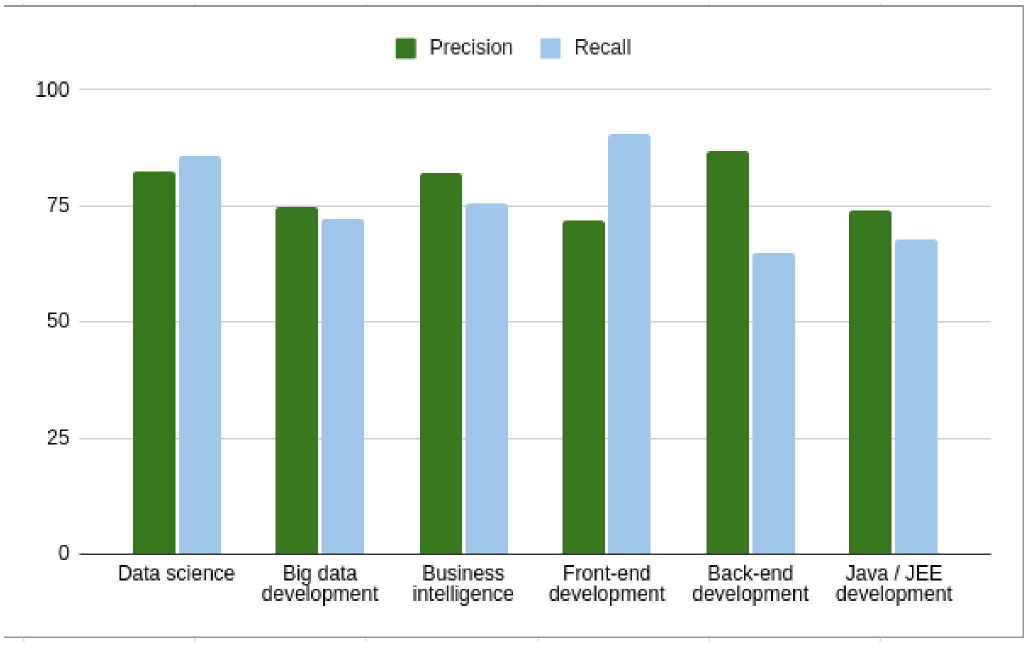
Table 4 CV Set

CV id	*Experience (years)*	*Job position fields / Occupied positions*
	5	Data scientist – Big data developer Business intelligence
	10+	Front-end / Back-end developer
	6	Data scientist – Big data developer Business intelligence
	1	Big data / Back-end developer Business intelligence
	6	Java JEE developer
	8	Data scientist – Business intelligence
	10+	Data scientist Back-end / Java JEE developer
	10+	Business intelligence
	10+	Front-end / Back-end developer
	10+	Java JEE developer
	5	Back-end / Front-end developer Java JEE developer
	2	Back-end Front-end developer
	2	Back-end developer
	3	Data scientist – Big data developer Business intelligence
	6	Data scientist – Big data developer Business intelligence Back-end / Java JEE developer
	1	Back-end / Front-end developer
	3	Data scientist – Big data developer
	10+	Java JEE developer
	10+	Back-end / Java JEE developer
	7	Data scientist – Big data developer Business intelligence
	2	Back-end / Front-end developer Java JEE developer
	1	Data scientist – Big data developer Business intelligence
	7	Back-end / Front-end developer
	10+	Big data developer
	10+	Back-end / Front-end developer Java / JEE developer
	1	Back-end / Front-end developer

Table 5 Evaluating the system results

Job position field	*Average precision*	*Average recall*
Data science	82.43	85.78
Big data development	74.88	72.34
Business intelligence	81.98	75.49
Front-end development	71.79	90.51
Back-end development	86.72	64.81
Java / JEE development	73.86	67.9
Total average	78.61	76.14

Figure 8 Precision and recall



It is fair to mention that the job requirements specified by the recruiter, which could not be compatible with the job position, have a direct impact on the results of our proposed method. Hence, it needs to be done carefully.

5.4 *Results of the job offer classification according to the expertise level*

In this section, we present the data set used for the job classification according to the expertise level using the required experience. Moreover, we discuss the performance of the classifiers. Figure 9 shows the data set used in the job offers classification. Since the data set is noisy, and to have accurate metrics, we shuffle the data 100 times to obtain the training and the test data. We save the metrics for the classifier in each time, and we compute their mean.

Table 6 describes the resulted metrics for each classifier and shows the performance of each one. The results confirm that the Random Forest Classifier works better compared to the other ones. Thus we will use it to predict the new job offers and CV classes in 'model 2'.

Figure 9 The data set used for job offers classification according to the required expertise using the required experience

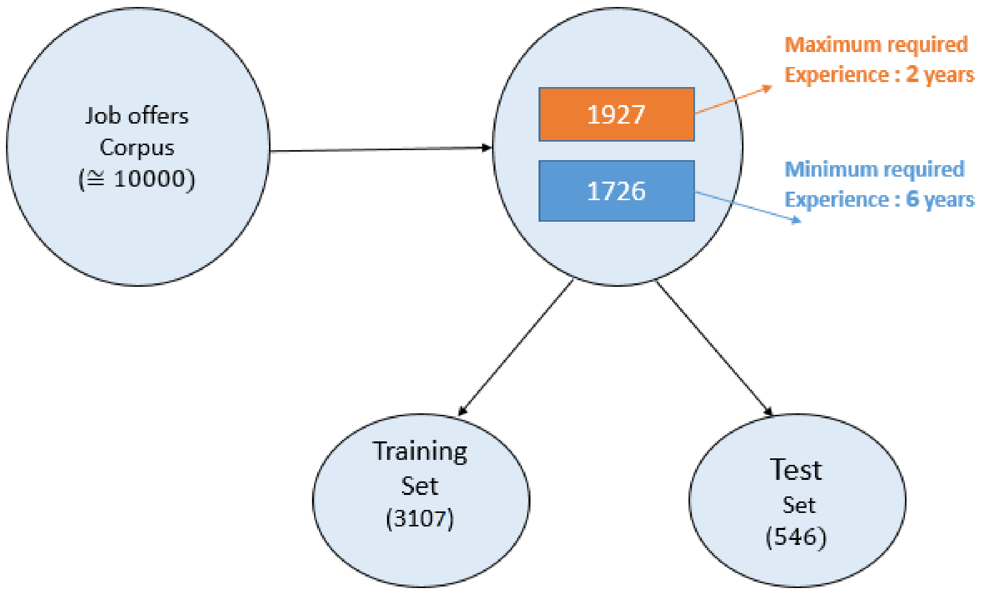

Job offers Corpus (≅ 10000)
1927
1726
Maximum required Experience : 2 years
Minimum required Experience : 6 years
Training Set (3107)
Test Set (546)

Table 6 Comparison table for performances of the classifiers models

Classifier	*Accuracy %*	*Recall %*	*F-score %*
Random Forest	81.2	81.08	81.14
XGboost	78.53	78.42	78.47
SVM	78.28	78.13	78.2
KNN	77.31	77.15	77.23
multiNomial Native Bayes	76.6	76.23	76.41
Decision Tree	75.33	75.21	75.27

5.5 *Results of the job offer classification according to the expertise level using activity verbs*

In Table 7, we compare the predicted classes of the CV classification method according to the expertise degree with the assigned ones. Since the predicted classes are ordered according to values of the fuzzy inference outputs, we replace the labels (‘A’–‘E’) by natural numbers to measure the forecast error. We used the real candidate work experience to assign the classes to be able to compute the forecast error of the system prediction. Thus we divided the work experience interval ([1, 10]) to five sets: [1, 2, 3, 4, 5] the same as the number of the predicted classes. To measure the effectiveness of our proposed method, we used the “mean absolute percentage error (MAPE),” which is a measure of prediction accuracy of a forecasting method, also used as a loss function for regression problems in machine learning. It usually expresses the accuracy as a ratio defined by the formula:

$$M = \frac{\sum_{i=1}^{n} \left| \frac{R_i - P_i}{R_i} \right|}{n} \tag{4}$$

where R_i is the real value and P_i is the predicted value. The classification error obtained using the formula is *M*=29.54% (Accuracy: 70.46%) shows that we can rely on this method to predict the expertise level especially for the candidate due to the challenge of extracting the work experience from the CV. Thus we could treat the overfitting and the underfitting issue.

Table 7 CV classification according to expertise level using activity verbs

CV id	*Work experience*	*Assigned class*	*Predicted class*	*Forecast error*
1	5	3	4	1
2	10+	5	1	4
3	6	3	4	1
4	1	1	1	0
5	6	3	3	0
6	8	4	3	1
7	10+	5	3	2
8	10+	5	4	1
9	10+	5	1	4
10	10+	5	4	1
11	5	3	5	2
12	2	1	1	0
13	2	1	1	0
14	3	2	1	1
15	6	3	3	0
16	1	1	1	0
17	3	2	4	2
18	10+	5	4	1
19	10+	5	4	1
20	7	4	1	3
21	2	1	1	0
22	1	1	1	0
23	7	4	3	1
24	10+	5	3	2
25	10+	5	3	2
26	1	1	1	0

6 Conclusion and future work

In this paper, we proposed an information retrieval system to automate the recruitment process in ICT domain. We aimed to enhance the matching between the candidates' CVs and the recruiters' job offers. We used an oriented ICT ontology to extract and infer text features from the recruitment documents. Fuzzy logic played an important role in metadata creation and weighting due to its ability to deal with qualitative parameters. We also used fuzzy logic with supervised learning in the CV/job offers classification task regarding their expertise level.

The experimental results have shown that our system provides effective results in terms of information retrieval evaluation measures. We described well the data used for tests in the test section, to show the efficiency of our system matching comparing it with the supervised one. Furthermore, we compared the candidates' classification results according to their expertise level with their work experience, to prove the effectiveness of our system to prevent over-qualification and under-qualification. The results show that the predicted classes scale well with the work experience.

For future work, we plan to propose a solution for CV/job offers clustering to make our system able to handle the massive amount of data along with its exponential growth in the job boards. We also plan to improve our matching by adding new features such as soft-skills to deal with personal traits and candidate personal information. Thus we can predict the candidate decision in the post-matching.

References

Al-Ramadin, T.A. and Al-ksasbeh, M. et al. (2015) 'A new approach for automated job search using information retrieval', *Global Journal of Computer Science and Technology*, Vol. 15, pp.1–8.

Balas-Timar, D. and Ignat, S. (2015) 'Conceptual applicant screening model with fuzzy logic in industrial organizational contexts', *Procedia-Social and Behavioral Sciences*, Vol. 203, pp.257–263.

Bourse, M. et al. (2002) 'CommOnCV: modelling the competencies underlying a curriculum vitae', *Proceedings of the 14th international conference on Software Engineering and Knowledge Engineering (SEKE'02)*, pp.65–73.

Brill, E. (1992) 'A simple rule-based part of speech tagger', *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Association for Computational Linguistics, pp.152–155.

Cabrera-Diego, L.A. et al. (2019) 'Ranking résumés automatically using only résumés: a method free of job offers', *Expert Systems with Applications*, Vol. 123, pp.91–107.

Catherine, R. et al. (Jan. 2010) 'PROSPECT: a system for screening candidates for recruitment', *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pp.659–668.

Converse, P.D. et al. (2004) 'Matching individuals to occupation using abilities and the O*NET: issues and an application in career guidance', *Personnel Psychology*, Vol. 57, No. 2, pp.451–487.

Desmontils, E., Jacquin, C. and Morin, E. (2002) 'Indexation sémantique de documents sur le Web: application aux resources humanise', *Proceedings of Journées de l'AS-CNRS Web sémantique*, pp.1–5.

Dubois, D. and Prade, H. (2005) 'Fuzzy elements in a fuzzy set', *International Journal Systems Science*, pp.1–6.

Faliagka, E. et al. (2012) 'Application of machine learning algorithms to an online recruitment system', *Proceedings of the 7th International Conference on Internet and Web Applications and Services*, pp.216–220.

Fensel, D. et al. (2000) 'OIL in a nutshell', *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, Springer, pp.1–16.

Gimpel, K. et al. (2010) *Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments*, Technical Report, Carnegie-Mellon University Pittsburgh Pa School of Computer Science.

Goswami, S. and Shishodia, M.S. (2013) 'A fuzzy based approach to text mining and document clustering', *arXiv preprintarXiv:1306.4633*.

Janusz, A. et al. (2018) 'How to match jobs and candidates-a recruitment support system based on feature engineering and advanced analytics', *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, pp.503–514.

Kannan, S. and Gurusamy, V. (2014) 'Preprocessing techniques for text mining', *Conference Paper (Multilingual Natural Language Processing)*, India.

Khandelwal, A. and Agrawal, A. (2015) 'An amalgamated approach of fuzzy logic and genetic algorithm for better recruitment process', *Transactions on Networks and Communications*, Vol. 3, No. 3, pp.23–23.

Khurana, D. et al. (2017) 'Natural language processing: state of the art, current trends and challenges', *arXiv preprint arXiv:1708.05148*.

Klement, E.P. and Mesiar, R. (2005) *Logical, Algebraic, Analytic and Probabilistic Aspects of Triangular Norms*, Elsevier.

Klosowski, G., Gola, A. and Świć, A. (2016) 'Application of fuzzy logic in assigning workers to production tasks', *Proceedings of the 13th International Conference on Distributed Computing and Artificial Intelligence*, Springer, pp.505–513.

Kuck, G. (2004) 'Tim Berners-Lee's semantic web', *SA Journal of Information Management*, Vol. 6, No. 1. Doi: 10.4102/sajim.v6i1.297.

Lassila, O. and Swick, R.R. (1999) *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation 22 February 1999.

Miller, G.A et al. (1993) 'Introduction to WordNet: an on-line lexical database (revised)', *International Journal of Lexicography*, Vol. 3, pp.1–87.

Roussey, C. et al. (2011) 'An introduction to ontologies and ontology engineering', *Ontologies in Urban Development Projects*, Springer, pp.9–38.

Samaila, V.D., Gumpy, J.M. and Manga, I. (2018) 'Application of fuzzy logic for personnel selection', *International Journal of Computer Applications*, Vol. 179, No. 15, pp.21–25.

Santos, C.D and Zadrozny, B. (2014) 'Learning character-level representations for part-of-speech tagging', *Proceedings of the 31st International Conference on Machine Learning (ICML'14)*, pp.1818–1826.

Toutanova, K.N. and Johnson, M.E. (2012) *Semi-supervised Part-of-Speech Tagging*, US Patent 8,275,607.

Van Belle, A., Dehling, E. and Foster, D. (2018) *Improving Candidate to Job Matching with Machine Learning*, Textkernel, Amsterdam, the Netherlands. Available online at: http://www.textkernel.com.

Voutilainen, A. (2003) 'Part-of-speech tagging', *The Oxford Handbook of Computational Linguistics*, pp.219–232.

Wang, C. (2015) *A Study of Membership Functions on Mamdani-Type Fuzzy Inference System for Industrial Decision-Making*, Thesis.

Yahiaoui, L., Boufaida, Z. and Prie, Y. (2006) 'Semantic annotation of documents applied to e-recruitment', *Semantic Web Applications and Perspectives*, *Proceedings of the 3rd Italian Semantic Web Workshop, Scuola Normale Superiore*, Pisa, Italy.

Zhihong, D., Shiwei, T., Ming, Z., Dongqing, Y. and Jie, C. (2002) 'Overview of ontology', *Acta Scicentiarum Naturalum Universitis Pekinesis*, Vol. 38, No. 5, pp.730–738.

Zwick, R. (1993) 'Hans-Jürgen Zimmermann: fuzzy set theory and its applications (2d ed.), (Book Review)', *American Journal of Psychology*, Vol. 106, No. 2, p.304.