DEEP, a methodology for entity extraction using organizational patterns: application to job offers

Halima Ramdani, Armelle Brun, Eric Bonjour, Davy Monticolo

Highlights

DEEP, a methodology for entity extraction using organizational patterns: application to job offers

Halima Ramdani,Armelle Brun,Eric Bonjour,Davy Monticolo

- Improving entity extraction in plain text documents using sequence labelling
- Proposing a methodology for creating an entity extraction corpus for automatic sequence labelling
- Using organizational patterns to address vocabulary evolution and ambiguity
- Evaluating the methodology on a real corpus of job offers

DEEP, a methodology for entity extraction using organizational patterns: application to job offers

Halima Ramdani[a,b,c,*], Armelle Brun[b], Eric Bonjour[a] and Davy Monticolo[a]

[a]*Equipe de Recherche sur les Processus Innovatifs, Lorraine University, ERPI, France*
[b]*Lorraine University, CNRS, Loria, France*
[c]*Xtramile, 11 rempart Saint-Thiebaut, 57000, Metz, France*

ARTICLE INFO

Keywords:
Entity extraction
Organizational patterns
Methodology
Sequence labelling
Recurrent neural network
Job offer entity extraction.

ABSTRACT

Plain texts written in natural language may have several specific features, such as organizational patterns and an ambiguous and evolving vocabulary. From the literature, entity extraction approaches are not sufficient to consider these specific features jointly. To address this issue, we propose DEEP, a methodology that improves the quality of entity extraction by using organizational patterns through a sequence labelling technique. To this end, DEEP creates a high-quality corpus and relies on an appropriate learning algorithm. DEEP is validated on a real corpus of job offers. Experiments show that (1) considering organizational patterns improves the quality of entity extraction, (2) vocabulary evolution is taken into consideration and ambiguity in vocabulary is reduced, (3) DEEP provides clear guidelines for the creation of a high-quality corpus for entity extraction, (4) the Bidirectional Long Short-Term Memory + Conditional Random Field architecture for sequence labelling is the one that takes the most advantage of the organizational patterns.

1. Introduction

Plain text documents are a vital and rapidly growing part of online information in various domains, e.g., in biomedical, research papers, and recruitment [22, 43, 35]. A single plain text can contain as much information as a small, structured database. Figure A.1(a) gives an illustration of a plain text in the case of job offers. Entity extraction expedites a search process in a text by identifying, extracting and determining all the appropriate tags for words or series of words in that text (segment). This task is critical, especially for plain texts written in natural language, and is increasingly concerned with the specific features of natural language processing, including: 1) vocabulary ambiguity, 2) vocabulary evolution, 3) spelling mistakes, etc. [41]. The literature on entity extraction approaches for natural language plain texts generally considers a subset of these specific features, but not all at once [6, 14, 9]. Besides, most plain texts written in natural language have organizational patterns. This refers to the organization of ideas in a text [8] by specifying the logical connections among ideas and the subordination of some ideas to others [33]. Moreover, it provides an organizational layout that not only guides the author when writing but also helps the reader identify interrelationships among ideas in the text. Organizational patterns for text are also referred to as text structure [42] and can be employed as a useful strategy to develop readers' comprehension [42, 8].

Sequence labelling, such as part-of-speech tagging, semantic annotation, etc., has long been of particular interest for natural language processing [39, 24, 13], due to its ability to consider the series of words as a sequence, and the series of sequences as the text structure. The sequence labelling task is an entity extraction approach that assigns a categorical label to each sequence by considering a text as a sequence of semantic words [30]. For example, in cooking recipes, the typical labelling of the text "Cut lengthwise half of the banana. If you are brave, you can also cut slices" could assign the label "ingredient" to the sequence "banana", the label "quantity" to the sequence "half" and the label "action" to both sequences "cut lengthwise" and "cut slices". Automatic sequence labelling is an entity extraction task that requires an annotated corpus [29] (labelled sequences from a plain text corpus) that plays the role of a learning corpus. Indeed, the sequence labelling task for entity extraction runs a supervised learning algorithm to further automatize this task. The learning corpus, including the labelling used, is essential and influences the quality of sequence labelling [15, 38].

The labelling of the learning corpus is typically performed manually by experts in the field [20, 11]. Yet, this task is costly and time-consuming, and it is constrained by the need for such experts, who may not be available. To make the corpus labelling task more accessible, some web platforms (Kaggle[1], Data gouv[2], etc.) propose ready-to-use corpora for sequence labelling.

However, there are some drawbacks to using such corpora: 1) they represent specific domains, 2) they are not customizable for a particular need, 3) the corpus size may not be sufficient for a particular need. In some cases [36], crowdsourcing can be used to create corpora, for instance, Amazon Mechanical Turk[3]. Crowdsourcing is a participatory online activity in which an individual or company proposes a group of people with diverse knowledge and backgrounds to perform a task on a voluntary basis [36]. Nevertheless, the difficulty comes from having the same annotation between

[1] https://www.kaggle.com/datasets

[2] https://www.data.gouv.fr/fr/datasets/all-datasets/

[3] https://www.mturk.com/

annotators and avoiding subjectivity or bias in understanding the plain text. These customization and crowdsourcing tasks require making clear guidelines that can be understood and implemented regardless of how difficult the task is and the area of expertise of everyone, especially for texts written in natural language. The objective of our work is to design a semi-automatic entity extraction approach on plain texts written in natural language by using the organizational patterns. For this purpose, we propose a dedicated methodology named DEEP, a methoDology for Entity Extraction using organizational Patterns. This approach consists of two main steps: (1) a manual corpus labelling for entity extraction that provides clear guidelines, i.e., that contains as few ambiguities and as little subjectivity as possible, and a manual annotation using the guidelines, (2) an automatic entity extraction. This article validates the hypothesis that sequence labelling is an approach that can take advantage of organizational patterns (H), while taking vocabulary evolution and ambiguity into consideration and improving the quality of automatic entity extraction.

In summary, the contributions of this paper are as follows:

- Design a methodology for the manual creation of a high-quality annotated corpus that reduces the required time for the expert by leveraging their domain knowledge to create clear guidelines for annotators and that contributes to reducing manual annotation uncertainties, thus improving the quality of entity extraction.
- Consider the organizational patterns in plain texts to improve the automatic entity extraction task.
- Automatically extract entities on job offers for an application to e-recruitment.

The paper is organized as follows. Section 2 explores the approaches used in the literature for entity extraction. Three approaches are compared: rule-based, ontology, and sequence labelling. Section 3 defines the methodology for entity extraction in plain text. Section 4 presents the experimentation carried out on a real corpus of job offers. In this section, we validate the hypothesis of our work. Section 5 is devoted to the conclusion and research perspectives.

2. Literature review

This section is dedicated to reviewing the main approaches used in the literature for plain text entity extraction. These approaches aim to associate an entity with a sequence of words. For the rest of the paper, the label/sequence pairs represent the entity and sequence of words. We specifically focus on three approaches as they are the most popular in the literature: rule-based, ontology-based, and sequence labelling. We will mainly go into more detail about the sequence labelling approach as it is the most promising strategy in the literature and we will explain why we chose it over the alternative approaches.

2.1. The rule-based approach

The rule-based approach is used in a wide range of application domains, including recruitment [7] and health [6]. Three rule-based techniques are classically used: regular expressions, dictionary of words, rules entity extraction.

Regular expressions consider a word or a set of words as a sequence of characters that define a search pattern. This search pattern is used to associate the matched search patterns with a label [7].

Dictionary of words is an enumeration of words associated with a label. For example, the words "full-time" and "part-time" in job offers are associated with the "contract type" label [7].

Rules entity extraction is a technique that considers the organizational patterns of the plain text to generate label/sequence pairs by associating a word or set of words with a predefined label [18]. An example of a rule could

be: "Each section begins with a title that defines its content".

However, the rule-based approach is limited in that it cannot consider changes in vocabulary (if a word is not represented in the dictionary of words or through a regular expression, it cannot be extracted). Nor does it consider any ambiguity of the language (if a word or sequence can be associated with two different labels). Finally, the entity extraction rules cannot be applied to plain texts that contain implicit and changing organizational patterns. In the next section, we focus on the ontology-based approach.

2.2. The ontology-based approach

An ontology is an organized vocabulary representing knowledge of a domain [37]. This knowledge is often represented as a set of concepts and instances (values), organized hierarchically and structured by relationships. Entity extraction using an ontology consists of associating a sequence of words with one (or several) concept(s) (label). The literature relies on either an existing ontology [46] or a purpose-built ontology [14].

The ontologies can be applied to natural language and different organizational patterns. When a new concept appears, it is necessary to add it to the ontology. The cost associated with this is significant [1]. Consequently, the ontological approach does not allow automatic, straightforward management of vocabulary evolution.

2.3. The sequence labelling approach

2.3.1. Principle

The sequence labelling approach can be applied to any plain text written in natural language [17, 11]. Sequence labelling considers a plain text as an order list of sequences, where a sequence is a series of words. The goal is to assign one label per sequence [40]. In the example, "we are looking for a back-end developer for our application", the sequence "back-end developer" is labelled "position". The label/sequence pairs represent the entity/segment. This approach consists of three steps [28]: creating the learning corpus, enriching the learning corpus, applying the learning algorithm.

Creating the learning corpus is a step where the corpus plays the role of learning corpus, made up of a labelled dataset. The goal is to manually associate label/sequence pairs from plain text.

Enriching the learning corpus is a step that provides an enriched representation of the words. This step performs corpus data processing to retain useful information and obtain valuable data [17]. It transforms the data into a format that will be both more effective and adequately prepared. A plain text written in natural language often contains mistakes. The dataset needs to be pre-processed to create a more accurate, robust, and high-quality learning model. If the corpus data is not of good quality, ambiguity may occur, and misleading results may be obtained [12]. The text from the corpus needs to be tokenized and normalized. Tokenization splits long text strings into tokens (smaller pieces of words). Large sentences can be tokenized into words, and chunks of text can be tokenized into sentences, etc. Normalization refers to a series of tasks that makes all texts equal: converting all texts to the same case (upper or lower), converting numbers to their word equivalents, etc. Normalization allows processing to proceed uniformly. Stemming and lemmatization are commonly used for the normalization task [44]. This step also aims to add supplementary information, such as morphosyntactic labelling (grammatical tagging of the words) [4], estimation of the position of the word in a sentence, or the position of the sequence in the text [4]. This information helps to consider the word context in the plain text and therefore maximizes consideration of any change in vocabulary.

Applying the learning algorithm is the step that allows the sequences and associated labels of a plain text to be learned, to label new documents automatically [21]. This algorithm is used on the enriched corpus.

2.3.2. Choice of learning algorithms

Sequence labelling is often approached in the literature as a supervised learning problem. Many works address this problem by using support vector machines (SVM) [5], or recurrent neural networks (RNN) [11]. SVMs are chosen for their simplicity. They are used, for example, to classify information [25] (postal code, salary, etc.) from a job offer. However, they do not make use of any information about the organizational patterns of the plain text or the context of each sentence. To address the challenge of capturing long-term dependencies between sequences, and thus the organizational patterns of the plain texts, some works have focused on Long Short-Term Memory (LSTM), a network of recurrent neurons with short and long-term memory. The ability of RNN to process sequences in both directions (bi-LSTM) (left-to-right and right-to-left chaining) [27] is advantageous [23, 19]. Using such a bidirectional feature allows to run the input sequences in two ways: from past to future and from future to past. The model thus preserves information from both the past and the future, considering the order of sequences. More recently, some works have used CRF (Conditional Random Field) algorithms [26]. CRF captures the dependencies between predicted labels at different steps of learning by analysing the probabilities of transition from one step to another. Also, several works have used CRF coupled with LSTMs, and Bi-LSTMs [31] for their adaptability to sequence labelling. CRF algorithms combined with RNN have allowed more accurate and efficient modelling of labels [11] compared to other learning methods, such as a simple RNN or SVM. In this work, we examine whether they maintain their advantage over other algorithms in the case of plain texts having organizational patterns as well as when considering vocabulary evolution.

2.4. Synthesis

The literature review has shown that the rule-based and ontology-based approaches are useful since they rely on predefined information, such as search patterns, dictionaries or rules. However, they cannot address vocabulary evolution, and they do not (and cannot) use the organizational patterns of the plain text to improve the entity extraction quality. These approaches need recurrent iterations over time to consider new words, new sentences, or changing organizational patterns. In contrast, sequence labelling is an approach that can consider the organizational patterns and vocabulary evolution if the learning corpus is well created and if the learning algorithm is well chosen. Nevertheless, if the corpus used for automatic sequence labelling is not of high quality, misleading results may be obtained [12].

To overcome this limitation, our work mainly focuses on designing a methodology for entity extraction with the goal of being used in sequence labelling.

3. DEEP, a methodology for entity extraction using organizational patterns

In this section, we present DEEP, a semi-automatic methodology for entity extraction using organizational patterns.

3.1. Overview of the steps of DEEP

DEEP intends to build a semi-automatic entity extraction corpus to avoid problems in connection with the use of an existing corpus and issues due to the presence of uncertainties that occur when annotating a corpus manually. DEEP overcomes the limits related to the following specific features: (C1) variability and uncertainty of manual sequence labelling, especially due to natural language characteristics; (C2) ambiguity between the label/sequence pairs (uncertainties when choosing between two labels for a sequence); (C3) vocabulary evolution. DEEP follows two main processes: (A1) manual corpus labelling for entity extraction and (A2) automatic entity extraction.

DEEP is modelled as an activity diagram following the traditional way of representing a series of actions or a flow of control in a system. The structure of the proposed methodology is detailed in Figure 1, where A_x denotes process x, A_{xy} denotes activity y of process x and A_{xyz} denotes sub-activity z.

3.2. The actors involved

The DEEP methodology involves four main actors with different expertise:

The master manages the methodology implementation. She/he is familiar with the application domain and has the required knowledge and understanding of its essential aspects.

The expert has domain expertise that includes knowledge and understanding of the essential aspects of a specific domain of inquiry [32]. The expert's time is restricted, and they cannot devote much time to the annotations.

The annotator is a person who is not specifically familiar with the domain. This actor annotates documents manually following the predefined guidelines.

The machine learning developer is a person who has the capabilities to choose and apply a learning algorithm to an annotated corpus.

Involving these four actors allows activities to be performed in parallel, thus reducing the time required by DEEP. Experts are often busy, come with high labour costs and are involved in tasks that require a high level of expertise. Annotators (possibly a large number) who are not familiar with the domain are involved to facilitate and improve the speed of the methodology's implementation (while maintaining the quality of annotation), as their task can be conducted in parallel. That being said, it is also possible to have only one person who carries out all the activities, but the duration of the annotation work will be longer.

3.3. Semi-automatic entity extraction

DEEP uses a set of plain text documents (plain text corpus) written in natural language as its primary data source. This corpus is divided into two separate corpora:

- the validation corpus that is made up of x% of the initial plain text corpus. This corpus is used to prepare and validate the guidelines (that will be presented hereafter) for manual annotation. The validation corpus is also used to create the gold standard corpus, as we can see in Figure 2. The gold standard corpus (GSC) is a corpus annotated by one or multiple experts. The GSC is intended to be a reference corpus to evaluate the annotations made by non-expert annotators.
- the rest of the plain text corpus ($100 - x$%). This corpus is annotated following the final guidelines.

3.3.1. Manual corpus labelling for entity extraction (A1)

In the proposed methodology, the first step is to create a corpus composed of label/sequence pairs from the plain texts through a manual corpus labelling of the plain texts. The main strength of DEEP falls within this step, which is carried out by the following actors: the master, the expert, and the annotators. This step is made up of three main activities (Creation of the guidelines for annotators (A11); Manual labelling of the validation corpus (A12); Manual annotation of the rest of the plain text corpus (A13)) that are described in this section and represented in Figure 1.

1. **Creation of the guidelines for annotators (A11)**

   A11 contributes to the development of an unambiguous method of manually labelling a corpus by establishing guidelines. The objective is to avoid any biases linked to the subjectivity of each annotator and ambiguities related to the organizational patterns of plain texts. Creating such guidelines is even more critical when the labelling of a corpus is distributed among several annotators. A11 is performed by both the master and expert and has six main sub-activities:

   (a) **Definition of the organizational patterns of the texts (A111)**

   A111 aims to define a model of organizational patterns. This activity requires knowledge of the domain and some years of experience to determine:

- **The most common order of information**. For example, in cooking recipes there is a sequential order of information. The writer follows a step-by-step pattern: starting with a list of ingredients and following it with a list of actions.
- **The different sections**. A plain text written in natural language is composed of different sections. It is represented by a series of semantic sections, with each section containing specific information types. For example, in a research article the sections are as follows: title, abstract, introduction, materials and methods, discussions, limitations, acknowledgments, and references. In a resume, the sections are identity, education, experiences, and hobbies.
- **The description**. A description of each section and of the type of organizational patterns is strongly recommended to describe the information content of each section.
- **The signal words.** These words help in identifying the particular type of organizational patterns. For example, in the introduction of a research paper the signal words may be: context, need, contribution, question, difficulty, problem, challenge, propose.

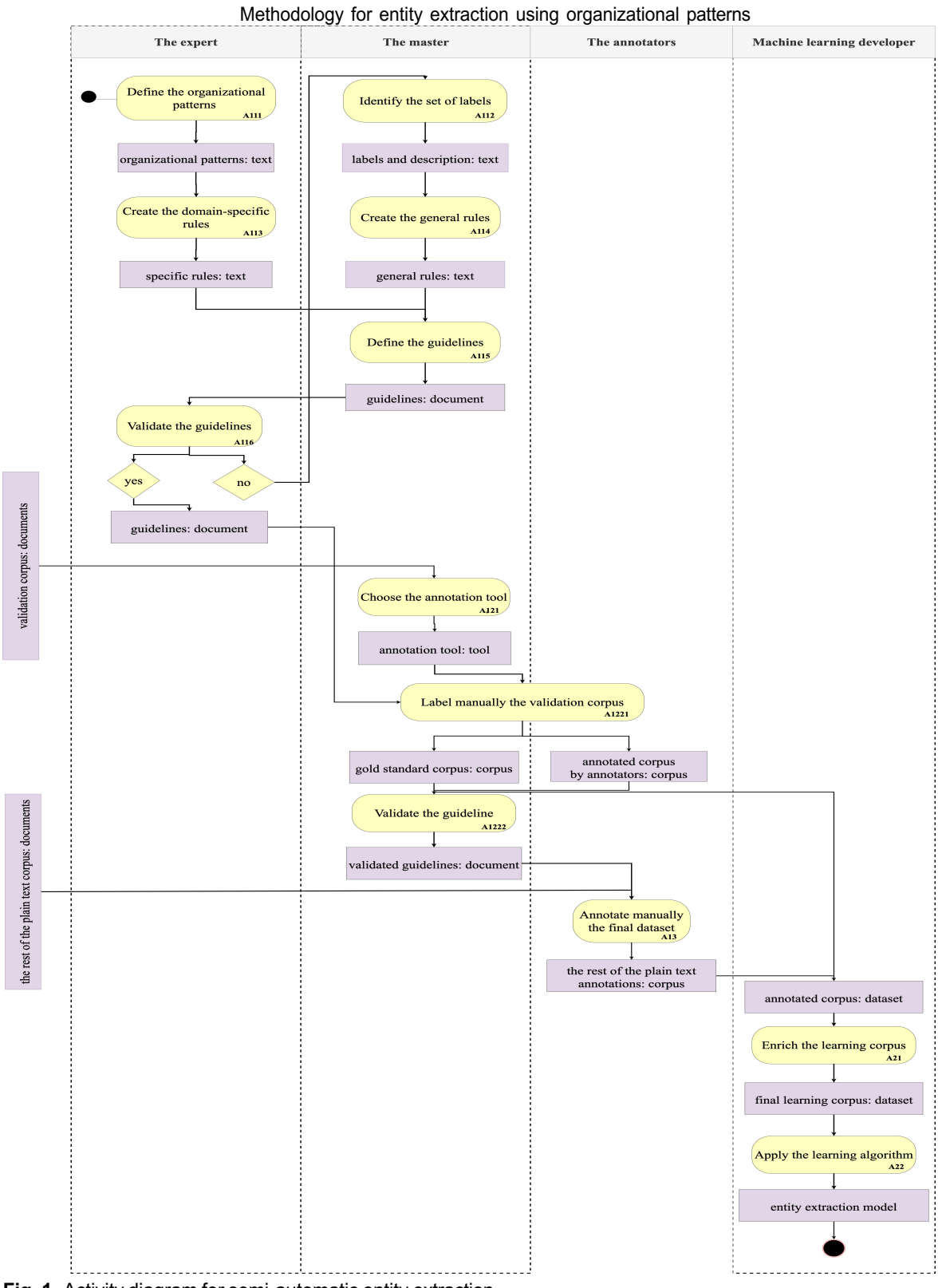

Fig. 1. Activity diagram for semi-automatic entity extraction.

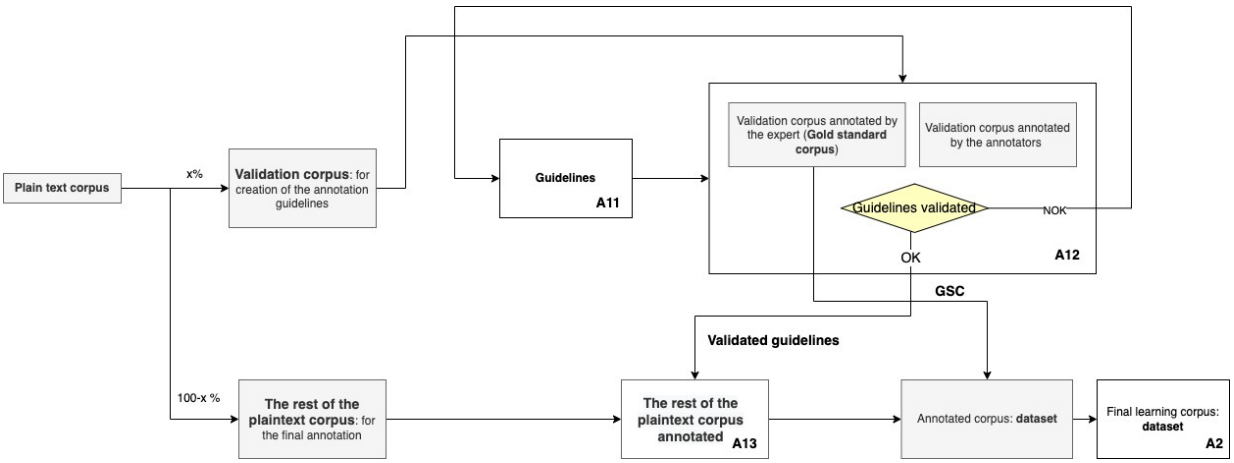

Fig. 2. Description of the data used in DEEP.

- **The sentence topics.** They describe the sequences that may be related to a label. For example, in an introduction, a sentence topic may be: "The main contribution is as follows".

A111 is therefore mainly conducted by the expert in interaction with the master. The expert proposes a typical organizational pattern that is the most common structure among plain texts. We assume that exploiting the organizational patterns has the advantage of limiting the ambiguities in the manual annotation, since each section represents a specific type of information. It also helps the annotators to familiarize themselves with these organizational patterns to save time while labelling. Furthermore, A111 identifies the organizational patterns, consequently allowing the consideration of vocabulary evolution, since it represents a series of sequences and words that can identify a particular label's position in the document. However, the organizational patterns depend on the author of the text. The model that is defined cannot be generalized across all plain texts since the content and the sequence order may differ. As a result, the uncertain organizational patterns will require consideration in the plain text's heterogeneous annotation. In the case of uncertainties, the expert highlights the section order that can generate uncertainties (while annotating for non-expert people), to consider them in the next activities as rules to update the guidelines.

(b) **Identification of the set of labels (A112)**

A112 aims to identify the set of labels useful for entity extraction, their specificities, and descriptions. We assume that the list of labels is finite and can be developed manually, based on a specific need. The sequence labelling approach allows the sequences associated with these labels to be identified automatically. A112 is carried out by the master, who has an overview of the needs. A112 helps the non-expert annotators familiarize themselves with the labels and associate the sections with the labels.

(c) **Creation of the domain-specific annotation rules (A113)**

A113 is relevant when there are instances of domain-specific rules. A113 is performed by the expert chosen for their expertise in the domain and builds up a detailed overview of the organizational patterns from A111. They create domain-specific rules that can help reduce the ambiguities in the uncertain structure when the organizational patterns that are chosen cannot be generalized across all the plain texts. Rule creation has the advantage of reducing ambiguities during the manual labelling in a specific domain. For example, in some cases, a sequence can contain two different labels. The "developer in Python"

sequence is an occupation but contains a hard skill. This sequence can generate uncertainties when choosing between the two labels. The associated rule could be "If the associated sequences contain a hard or soft skill, do not annotate it as a skill but as the label it is related to". This rule intends to prioritize the labels in the sequences. Another rule could be "If any of the words 'mandatory' or 'optional' appear after an 'experience' label, then the sequence is associated with the label 'experience'". Appendix B presents examples of rules and their logical statement.

(d) **Creation of general annotation rules (A114)**

A114 does not require much expertise in the domain and is therefore performed by the master. Creating general rules allow a common way for labelling a document and avoids ambiguities and subjectivity. The general annotation rules are classified into three categories:

Punctuation rules. The aim of this category is to create punctuation rules. For example, "Include in the annotation the punctuation in the sequences". This category of rules is important to ensure that the annotators work in the same way when annotating in order to avoid uncertainties.

Sentence connectors. This category of rules is used to assign a status to each label/sequence: mandatory or not. "Or" and "And" connectors are useful for creating a profile from a natural language text.

Natural language rules. This category aims to define specific rules for sequences that can be written in different ways according to the author of the document. For example, "Consider in the sequence associated with the label *Date* all the information of year, month, day, even if it is written in letters or units". However, the general rules can be exhaustive and depend on the annotators. Iterations may be needed to consider the particularities of the domain.

(e) **Definition of the guidelines (A115)**

A115 is managed by the master, who collects the outputs from the previous sub-activities and creates a document containing the guidelines.

(f) **Validation of the guidelines (A116)**

A116 can be managed by the expert. If the expert validates the guidelines, we can move on to the next activity A12. If not, there is a further iteration from A112 to A115.

2. **Manual labelling of the validation corpus (A12)**

A12 aims to further refine and validate the guidelines. The master (or the expert) annotates the validation corpus manually and produces the gold standard corpus (GSC) that will be considered as a reference. In turn, the annotators must manually annotate the validation corpus. The evaluation of the quality of the latter annotated corpora compared to the GSC will allow the master to validate the guidelines or to point out the need to refine them.

(a) **Choice of the annotation tool (A121)**

A121 aims to choose an annotation tool and is carried out by the master. Software exists to simplify annotations by proposing simple interfaces, possibly including an automatic pre-labelling phase, for example, Dataturks[4] or Gate[5]. These two software packages allow manual annotations by proposing an easy

[4] Annotation tool: https://dataturks.com/

[5] Annotation tool: https://gate.ac.uk/

interface for associating a label with a sequence.

(b) **Validation of the guidelines (A122)**

i. **Manual labelling of the validation corpus (A1221)**

A1221 is carried out by the annotators and the master. The master annotations will be used as the GSC to evaluate the annotators' annotations. In parallel, the annotators label the validation corpus, following the guidelines provided by A11.

ii. **Guidelines validation (A1222)**

A1222 is managed by the master. It is based on the evaluation of labelled corpora. Two measures are used: the inter-annotator agreement (IAA) [2], which has traditionally been used in the literature, and the gold annotator agreement (GAA) [16].

IAA measures how well several annotators make the same annotation decision (following the guidelines). In concrete terms, it represents the percentage of overlapping choices between the annotators. IAA represents the level of understanding and avoidance of ambiguities in the annotation. A low IAA value means that the annotators do not agree on the labels to be assigned to a sequence. This conclusion means that the rules provided in the guidelines may create ambiguity and need to be modified for that label. GAA is a measure that compares the overlap between the documents labelled by the annotators and the gold standard corpus (GSC) [34], which is a trustworthy annotated corpus. This measure is used to evaluate whether the domain-specific rules are properly defined in the guidelines. The GAA measure helps to save time creating the rules by identifying domain-specific ambiguities quickly. Depending on the IAA and GAA values, the master either validates or modifies the guidelines to address any ambiguities and misunderstandings. An IAA value greater than 79% (generally considered as a high value [3]) and a GAA value greater than the IAA indicate that the domain-specific rules in the guidelines are well defined, otherwise the master must modify it and go back to activity A112. The master identifies the sequences that do not overlap the gold corpus to create a specific rule to make this modification. If the threshold is exceeded, the master validates the guidelines.

3. **Manual annotation of the rest of the plain text corpus (A13)**

This final step aims to label the rest of the plain text corpus using the guidelines. As the guidelines have been validated, i.e., the IAA and GAA are maximal, each document can be annotated by one annotator only.

3.3.2. Automatic entity extraction (A2) using the annotated corpus

This step aims to use the annotated corpus provided by A1 to enrich this corpus (A21) and learn the entity extraction model (A22) to automatize the task of entity extraction. A2 has two main sub-activities:

1. **Enriching the learning corpus (A21)**

This sub-activity serves to enrich the annotated corpus by applying data processing to add lexical information. Syntactic disambiguation is used to determine the correct grammatical category of each word according to its context. This step also aims at vectorizing the sequences. Simple lexical vectorization may be used. On the other hand, to have better results and capture the semantic sense of sequences, a word embedding such as BERT could be used. It is a method of pre-training language representations (trained on a large text corpus (like Wikipedia). BERT is the first unsupervised, deeply bidirectional system for pre-training natural language processing [10]. By translating a word to an embedding, it becomes possible to model its semantic importance in

numeric form and thus perform mathematical operations on it.

2. **Learning the entity extraction model (A22)**

   A22 aims to apply a learning algorithm for automatic entity extraction. To do so, it exploits the final learning corpus provided by step A21. This step also aims to determine the inputs and outputs of the model and to define the methods for evaluating and validating the chosen algorithms. The evaluation technique proposed in this methodology is cross-validation. The training and test sets are randomly selected. The evaluation measures proposed are the conventional measures of precision (P), recall (R), and F1 score (F1).

4. Experiments and results

We propose evaluating DEEP in the domain of job offers. A job offer is a plain text with three main characteristics:

- It is written freely in a natural language. The information and the vocabulary used in a job offer may therefore differ between offers.

- It has organizational patterns. A job offer is a series of sections, each containing a specific type of information. For example, the "company description" section contains its name, values, and type of structure. Even though a typical order between these sections is commonly adopted, it may vary from one job to another.

- The vocabulary evolves. Over the last few years, we have seen the emergence of new jobs, new skills, therefore the vocabulary used in job offers tends to evolve. Furthermore, some information shares common vocabulary (for example, skills and experience). Therefore, it can lead to ambiguity when labelling the job offer.

4.1. Dataset

For the evaluation, we exploit a real corpus of job offers that we built up, composed of 3,335 French job offers, between 2017 and 2019 (1,094,562 words), extracted from several French job search sites (Indeed, Leboncoin). The job offers were evenly distributed between 25 sectors of activity (human resources, computer science, education) and each contained an average of 328 words with a standard deviation of 20.4. There were 21,790 different words in the corpus.

The dataset may seem small. However, the objective is to show that a high accuracy can be obtained even with a small number of documents when the learning corpus contains few ambiguities, and the chosen algorithm is adapted. The participants in this evaluation are: a recruiter in the role of the expert; a Human Resources employee with one year's experience and a master's degree in the role of the master; and three master's students from a different domain (e.g., chemistry, IT) as annotators for the guidelines' composition; ten annotators for the validation of the guidelines and three for the final annotations on the rest of the plain text corpus.

Please recall that DEEP relies on the creation of a corpus (A1). In practice, two corpora are created. The validation corpus is used to validate the guidelines and contains 335 full-text job offers (10% of the corpus).

The dataset (plain text corpus) contains 3,335 job offers. In this experimental validation, we first validate the initial challenge of this work, which is managing the variability and uncertainty. Then we analyse the performance of the automatic entity extraction by using the corpus created in the previous step.

4.2. Does DEEP contribute to reducing the variability and uncertainty of manual entity extraction?

To answer this question, we first apply each activity from A1 to our corpus. After that, we compare the time and quality of annotations with and without the guidelines.

4.2.1. Manual corpus labelling for entity extraction (A1)

Firstly, the expert studied and analysed the organizational patterns of the job offers from the validation corpus (A111), resulting in a table including all the information to describe organizational patterns.

The different sections in job offers are the company description, the job description, the desired profile, the conditions, and additional information. **The most common order of information** is described as follows: the first section generally presents the company: its activities, size, and values. The second section is generally dedicated to the job description: the main tasks, the contract type, the salary. The third section relates to the desired profile: the required hard skills, experience, education. The last section details the conditions (for example, start date) and finally additional information such as the email contact address. This order confirms that recruiters generally follow organizational patterns to describe a job offer. **The description** of each section and the type of organizational patterns have been added to the guidelines. For example, the profile description section contains the experience duration and is followed by a skill, an occupation or a domain. **The signal words** chosen for the required experience are experience, duration, year, domain. The recruiter provided signal words for each section and label. **The sentence topics** are an example of sentences. For job descriptions, a sentence topic is "The main task of the job is to conduct simulations with the ocean biogeochemistry model". The sentences for the label "missions" use topics that describe an action.

Activity A12 aims to identify the set of labels. It is conducted by the master. In this experiment, the set of labels are chosen according to two goals: (1) job offer dissemination on job sites, as illustrated in Figure A.1(b) (as the input for this task is an organized document such as an XML. The XML feed is appropriate for representing hierarchical information, as it is a file containing tags to categorize information [45]), (2) using the entity and the associated sequences to compute similarities with resumes. To do so, the master identifies the following set of 11 labels: "city", "contract type", "education", "experience", "experience duration", "hard skills", "soft skills", "main tasks", "occupation", "postal code", and "salary". Some of these labels are commonly used in the literature, and others, such as "main tasks" and "soft skills", were added to the set of labels. The "main tasks" are not often included in the literature, but this identifies cross-functional skills. Furthermore, soft skills (for example, teamwork, autonomy) are essential information commonly used by recruiters. Finally, the expert validated the guidelines (A116). The master compared different annotation tools to choose the one that is suitable for this task (A211). As we can see in Table 1, the master compared different characteristics. The tool chosen by the master is Dataturks as it is the one that meets all the characteristics. The master then created three different annotation projects that contain the validation corpus of job offers for each annotator.

4.2.2. Results

For the manual labelling validation, the annotators took as their input the validation corpus both in the tool and the guidelines to tag the documents manually. Appendix D shows an example of a job offer tagged manually by an annotator in Dataturks. In this example, the "web developer (HTML/CSS)" sequence is labelled as "occupation". Both the IAA and GAA measurements are examined after all annotators have finished annotating the corpus. The inter-annotator agreement (IAA) was equal to 88%. The gold-annotator agreement (GAA) was 75%. As a result of the high IAA value, the master partially validates the guidelines and general rules. The GAA is lower than the threshold. He concludes that the rules relating to the human resources domain are not clear. The sequences that did not match the gold standard corpus of job offers annotated by the expert are extracted. After studying these sequences, the master added four rules that were validated by the expert to reduce the ambiguity related to them. The second iteration of the validation which is dedicated to the validation of the new guidelines was conducted by an increased

Table 1
Annotation tools comparison.

Characteristics	Dataturks	Brat	Amazon Mechanical Turk	LabelBox
Open source	✓	✓	✓	✓
Web/Standalone	Web	Web/Standalone	Web	Web/Standalone
Collaborative annotation	✓	✓	✓	✓
Json/XML export	✓	✓	✓	✓
Filter by annotators	✓	x	✓	✓
Reports	✓	x		✓
Crowd-sourcing	✓	x	✓	✓
User-friendly	**	*	*	***
Easy tagging (clear box limitation)	***	*	*	**
Easy access and project creation	**	*	**	**
Price	Free	Free	Free	Paid

Table 2
Report of time spent for each annotator (seconds).

	First iteration (A221)	Second iteration (A222)	Final annotation (A3)
Number of documents	335	125	3,335
Annotators			
1	240	75	150
2	230	80	165
3	200	75	140

Table 3
Report of time spent for each annotator without/by executing activities A113 and A114 (in seconds).

	Annotation without A113 and A114 output	Annotation with A113 and A114 output
Number of documents	30	30
Annotators		
1	380	295
2	365	248
3	370	255
4	345	230
5	365	236
6	410	252
7	466	281
8	365	302
9	320	298
10	497	320
Average	388.3	271.7
Standard-deviation	51.86	29.69

number of annotators. A total of ten annotators took part in this validation process. This validation resulted in an IAA equal to 94% and a GAA equal to 89%. The standard deviation was 2.4%. Both the high GAA score and low standard deviation demonstrate that the guidelines are effective. The second iteration also required less time than the first, as seen in Table 2, which compares the time required to achieve an annotation in the first and second iterations by the three annotators who participated in both validations. Our experiment highlighted the fact that "A113. Creation of the domain-specific annotation rules", "A114. Creation of general annotation rules" and "A1221. Manual labelling of the

validation corpus" helps the annotators associate labels with the sequences, even for domain-specific vocabulary. For this purpose, we compared a manual annotation based on all the proposed activities and a manual annotation without sub-activities A113 and A114. As we can see in Table 3, the guidelines contribute to saving time, on average 116.6 seconds, which is 30% lower than the time spent without the guidelines in place. The standard deviation between the annotators is also 43% lower than the standard deviation between annotators without using guidelines. These results confirm the homogeneity between the annotators. Considering the challenge (C1) tackled in this work, we consider that DEEP can reduce the variability in manual annotations. In the next section, additional experiments will take place to confirm this conclusion by evaluating automatic entity extraction using the manual annotated corpus.

4.3. Is automatic entity extraction accurate?

For this question, we begin by applying the second step of DEEP on our dataset. To ensure the quality of the approach, we have applied different machine learning algorithms to choose the most accurate. We also compare our entity extraction model to the cutting-edge approaches presented in the background section.

4.3.1. Automatic entity extraction (A2)

In this activity, the corpus used is the output of the manual annotation from A21 (the final learning corpus). This step was carried out by three annotators, who each annotated one third of the corpus. The labelled corpus created after manual annotation consisted of 317,693 labelled sequences, representing 66% of the total. 34% of the sequences are annotated as "other" (not associated with any label) and therefore not essential (they do not provide important information). The pre-processing step was carried out on the final corpus using morphosyntactic analysis. We also added word lemmatization, the position of the sequence in the document, and the words composing it to the corpus. Appendix C shows the distribution of the different labels in the final corpus. We can see that “hard skills”, “main tasks” and “occupation” represent the most frequent labels in the offers. In addition, “soft skills” also represents an essential part of the labels. This information confirms that it is important information for the recruiter. Finally, we note that “salary” is the least-used label, which means that it is not mandatory information in the offers. These values will provide a basis for interpreting the results of the entity extraction model. For (A22), we opted to focus on the most popular models studied in the literature, which are CRF, LSTM CRF, and Bi-LSTM CRF. The input of these training models is the enriched corpus (the final learning corpus). The output of this entity extraction corresponds to the label/sequence pairs. The first experiment used the LSTM model with the same parameters as those used in [11], except for those that changed to adapt the model to the dataset size, average number of words per plain text and label: the embeddings used had a size of 150. The hidden layers of the recurrent neural network were 512 in size. Dropout [11], which is a regularization method to prevent overfitting during training, was 0.2. For the LSTM CRF and Bi-LSTM CRF models, we chose the same parameters as LSTM, to which we added an output layer. We are also interested in the SVM, which is widely used in the literature and has demonstrated outstanding performance. However, SVM does not consider the chain of sequences, i.e., the organizational patterns. SVM classification was chosen to validate the impact of considering the organizational patterns and, therefore, vocabulary evolution, reducing ambiguity associated with the shared vocabulary between labels, and the choice of the sequence labelling approach. To demonstrate the impact of organizational pattern, we have used the same word embedding as was used in the LSTM. The evaluation technique chosen was 10-fold cross-validation applied on the (enriched) final learning corpus. The dataset used is the entire corpus labelled (3,335 job offers).

4.3.2. Validation of (C1): variability and uncertainty of manual sequence labelling

Table 4 shows the precision, recall and F1 of the CRF, LSTM CRF, Bi-LSTM CRF and SVM for each label. We first observed that the average value of F1 for CRF, LSTM CRF, Bi-LSTM CRF was very high: above 0.89. We can

deduce that the sequence labelling approach is adapted to the entity extraction. We can also observe that the F1 values associated with each label for all the models are very high (at least 0.76). Note that the Bi-LSTM CRF model achieved the best performance on average. Also, the "salary", "postal code", "education" and "experience duration" labels were identified accurately by the three models with a minimum of 0.90, whereas the amount of training data on these labels was relatively low. The related sequences were easy to identify thanks to their commonly used position in the job offer. Moreover, regarding the "soft skills" label that the master proposed, its performance was particularly promising, especially for the Bi-LSTM CRF. It was the sixth-best label with a precision and recall score of 0.92 and 0.90, respectively. We also note that for the "city" and "occupation" labels, the Bi-LSTM model underperformed on recall compared to CRF and LSTM CRF. This can be explained by the fact that it requires more data due to its bidirectional feature. It should also be noted that in labelling the LSTM CRF and CRF models, the "experience" and "hard skills" labels are mixed in some job offers, as demonstrated in the "You have experience in the Python language" sequence. "Python language" can be considered either an experience or a skill. These two models did not distinguish between these two labels, unlike Bi-LSTM CRF, which had good recall on these labels, but lower precision. The bidirectionality of Bi- LSTM improves the consideration of plain text organizational patterns and thus favours the performance of this model. We can conclude that the methodology proposed for creating the corpus enables high-quality job offer entity extraction. Moreover, the model that provided the best performance on the job offers was Bi-LSTM CRF.

Table 4
Precision, recall and F1 of the supervised learning on the four models.

	CRF			LSTM CRF			Bi-LSTM CRF			SVM		
Measures	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1
Labels												
City	0.94	0.90	**0.92**	0.88	0.84	0.86	0.95	0.85	0.90	0.67	0.50	0.57
Contract type	0.92	0.85	0.89	0.90	0.91	**0.91**	0.96	0.87	**0.91**	0.90	0.46	0.63
Education	0.92	0.93	0.91	0.94	0.92	**0.95**	0.94	0.94	0.93	0.89	0.64	0.74
Experience	0.77	0.81	**0.91**	0.78	0.84	0.87	0.83	0.85	0.86	0.84	0.70	0.76
Exp. duration	0.97	0.96	**0.97**	0.96	0.96	0.96	0.99	0.99	**0.97**	0.60	0.36	0.45
Hard skills	0.84	0.75	0.80	0.77	0.76	0.76	0.76	0.86	**0.81**	0.69	0.90	0.78
Main tasks	0.86	0.90	0.88	0.91	0.87	0.89	0.93	0.91	**0.91**	0.82	0.64	0.72
Occupation	0.96	0.93	**0.95**	0.95	0.92	0.93	0.99	0.88	0.93	0.86	0.38	0.52
Postal code	0.99	0.99	**0.99**	1.00	0.95	0.97	1.00	0.97	**0.99**	0.83	0.69	0.75
Salary	0.93	0.91	**0.92**	0.93	0.90	0.91	0.91	0.93	**0.92**	0.88	0.80	0.83
Soft skills	0.91	0.87	0.89	0.88	0.79	0.89	0.92	0.90	**0.91**	0.77	0.60	0.68
Average	0.91	0.87	0.89	0.91	0.87	0.89	0.92	0.90	**0.91**	0.81	0.58	0.67

4.1. Discussion and hypothesis validation

Recall that this paper addresses the following hypothesis: (H) entity extraction quality can be improved by using organizational patterns. In addition, we assume that DEEP can manage the specific challenges: (C1) variability and uncertainty of manual sequence labelling, (C2) the ambiguity between the label/sequence pairs, and (C3) the robustness of vocabulary evolution. The sections hereafter are dedicated to discussing the validity of these hypotheses.

4.1.1. Validation of (H): using the organizational patterns improves the entity extraction quality

This article assumes that considering the organizational patterns could improve the entity extraction. To confirm

this hypothesis, we compared a Bi-LSTM learning model that is supposed to consider the organizational patterns (by considering the context of each labelled sequence) to SVM. SVM takes the sequences as input and the labels as output. It does not consider either the previous or the next labelled sequences to capture the context. Each of the sequences and corresponding labels are considered as independent from each other. Table 4 presents precision, recall and F1 obtained with SVM. We note a significant difference in the performance of SVM and Bi-LSTM. SVM has a mean precision of 0.81, which is significantly lower than that of Bi-LSTM CRF (0.92) and other algorithms (0.91). We can also note that for the label "hard skills", SVM generated fewer uncertainties than for the label "city". Indeed, since we may find the city of the job description and the city of the company head office, SVM barely differentiates between these two cities because of the unknown context, whereas Bi-LSTM is able to capture the context and therefore the structure. The significant performance of Bi-LSTM confirms that DEEP and the learning models can capture the organizational patterns of a plain text.

Table 5

Performance comparison between rule-based approaches using dictionary and Bi-LSTM CRF.

	Dictionary			Bi-LSTM CRF		
Measures Labels	P.	R.	F1	P.	R.	F1
Hard skills.	0.57	0.26	0.36	0.76	0.86	**0.81**
Main tasks	0.62	0.47	0.54	0.93	0.91	**0.91**

4.1.2. Validation of (C2): reducing the ambiguity between some label/sequence pairs

As mentioned in section 4, some job offers share common vocabulary (for example, for the "main tasks" and "hard skills" labels, the associated sequences can share a common vocabulary). This can therefore lead to uncertainties while annotating, and thus ambiguities in the learning model. To address this challenge, we first introduced the **"A111. Definition of the organizational patterns of texts"** sub-activity into the methodology to ensure that the annotators could easily associate certain labels with a sequence thanks to the organizational patterns. We noticed that annotating a job offer took 4 minutes when the annotator had the output of A111 at their disposal, against 7 minutes without. Moreover, including the **"A113. Creation of the domain-specific annotation rules"** sub-activity improved the annotation speed and score of our model.

As a second validation, two dictionaries of the most frequently used sequences related to "main tasks" and "hard skills" were created. The Bi-LSTM CRF model was compared to the "Dictionary of words" approach we replicated. The scores presented in Table 5 show a significant difference between the two approaches. We note a difference of more than 40% in terms of effectiveness. We can conclude that DEEP is able to reduce the ambiguity in our model compared to the rule-based approach.

4.1.3. Validation of (C3): robustness of vocabulary evolution

We assume in this paper that taking organizational patterns into consideration will help in evaluating vocabulary evolution. To assess the ability of DEEP to consider changes in vocabulary, we performed three types of evaluations:

- by applying the model on three unknown job sectors: public works; armed forces occupations; animal producers. The model gave an F1 score of 0.90, 0.89 and 0.89 for respectively 1%, 5% and 10% of the training size. The average score is 0.89.
- by replacing some known vocabulary (10% of the total number of words in the offers) in the test set with unknown vocabulary. We noticed that the labels associated with the sequences remained the same, even though

the vocabulary was unknown.

- by testing the model on one job offer dating back to 19 years ago. This offer contained four sequences that were not present in the supervised learning corpus. The decrease in the score can be partially attributed to the fact that organizational patterns have changed significantly since 2000. However, the sequences unknown to the system were labelled correctly. For example, the “strong entrepreneurial spirit” sequence was correctly labelled as “soft skills”.

Based on these experiments, we can confirm that DEEP indeed addresses vocabulary evolution efficiently.

5. Conclusion and further study

Entity extraction is used to associate an entity with a word or sequence of words in a plain text. This task is critical. Therefore, we propose a methodology for entity action for plain texts written in natural language by considering the organizational patterns, while managing both ambiguity and vocabulary evolution. Different approaches exist in the literature for entity extraction, but they do not jointly consider the specific features of documents written in natural language. In contrast, sequence labelling is an approach that is based on automatic learning from a labelled corpus.

The sequence labelling approach considers the context and can therefore consider vocabulary ambiguity and changes. Nevertheless, the quality of this approach is directly impacted by the quality of the corpus. This methodology helps to create a high-quality sequence labelling corpus and involves four main actors with different expertise. DEEP was validated on a real corpus of job offers. Its implementation in the job offers domain, with one expert, one master, and three annotators, took about 300 hours. Compared to the implementation time required by other automatic approaches, DEEP may appear more time-consuming. Nevertheless, this time is for the benefit of entity extraction and for the ability to tackle the different specific features: (C1) variability and uncertainty of manual labelling, (C2) improving the ambiguity between certain label/sequence pairs and (C3) considering vocabulary evolution. The different experiments showed that the "A113. Creation of the domain-specific annotation rules", "A114. Creation of general annotation rules" and "A12. Manual labelling of the corpus" sub-activities in the methodology are essential to avoid uncertainties while annotating and to address challenge (C1). For challenge (C2), the "A113. Creation of the domain-specific annotation rules" and "A111. Definition of the organizational patterns of the texts" sub-activities allow the annotator to have fewer uncertainties between the labels that share common vocabulary. Finally, for challenge (C3), the experiments show that considering the text structure can help to associate vocabulary that is unknown to the model with the right labels.

For automatic entity extraction, we compared different algorithm and entity extraction approaches from the literature. The algorithm that allowed us to have the best results (higher F1) was the Bi-LSTM combined with CRF. This architecture of recurrent neural networks demonstrated its capabilities to take context into consideration and make use of organizational patterns to associate sequences with labels. We also compared the proposed approach to the state-of- the-art approaches. According to the experiments, our approach is more effective in terms of vocabulary evolution and in reducing ambiguity between vocabulary.

As a result, DEEP does not require continuous updating of all new vocabulary, unlike other approaches, which therefore saves time. Notice that a smaller corpus can be labelled manually. Then, a semi-automatic annotation can be done, which results in a decrease in the time required. This research has practical and theoretical implications. From a theoretical point of view, it contributes significantly to the development of the entity extraction domain, by (1) proposing a multi-stakeholder methodology designed for entity extraction, (2) defining annotation guidelines that involve only non-expert annotators, with the following dual objective: time saving and a guaranteed high-quality manual annotation, (3) managing organizational patterns which improve the quality of automatic entity extraction,

(4) automatically extracting entities on job offers for an application to e-recruitment. From a practical point of view, the methodology is generic. It can therefore be applied to many fields, e.g., news, biomedicine, administration. Once the guidelines are established, it can be used by any non-expert annotator. In addition, the annotation step can be parallelized between several annotators while guaranteeing the consistency between them. It can thus be used on large corpora.

In future work, DEEP will be evaluated on annotators using crowdsourcing tools, to confirm its effectiveness with unknown annotators. We will also seek to improve entity extraction by including entity relation and active learning to improve the model. Furthermore, as DEEP is a generic methodology, it can be used for other purposes, such as identifying organizational patterns in documents or improving the evaluation of plain texts' similarity by exploiting their organizational patterns. As another perspective of research, we seek to semi-automatize the first activity of the manual annotation, by using the learning model to pre-annotate the plain texts and reduce the time consumed while annotating.

CRediT authorship contribution statement

Halima Ramdani: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing - Original Draft, Writing - Review & Editing. **Armelle Brun:** Conceptualization, Methodology, Software, Validation, Writing - Review & Editing Supervision. **Eric Bonjour:** Conceptualization, Methodology, Writing - Review & Editing, Visualization, Supervision. **Davy Monticolo:** Conceptualization, Methodology, Writing - Review & Editing, Project administration, Funding acquisition.

References

[1] Al-Aswadi, F.N., Chan, H.Y., Gan, K.H., 2020. Automatic ontology construction from text: a review from shallow to deep learning trend. Artificial Intelligence Review 53, 3901–3928. doi:10.1007/s10462-019-09782-9.

[2] Artstein, R., 2017. Inter-annotator agreement, in: Ide, N., Pustejovsky, J. (Eds.), Handbook of Linguistic Annotation, Springer Netherlands, Dordrecht. pp. 297–313. doi:10.1007/978-94-024-0881-2_11.

[3] Bobicev, V., Sokolova, M., 2017. Inter-annotator agreement in sentiment analysis: Machine learning perspective, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, INCOMA Ltd., Varna, Bulgaria. pp. 97–102. doi:10.26615/978-954-452-049-6_015.

[4] Bohnet, B., McDonald, R., Simões, G., Andor, D., Pitler, E., Maynez, J., 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia. pp. 2642–2652. doi:10.18653/v1/P18-1246.

[5] Bordes, A., Usunier, N., Bottou, L., 2008. Sequence labelling svms trained in one pass, in: Daelemans, W., Goethals, B., Morik, K. (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 146–161. doi:10.1007/978-3-540-87479-9_28.

[6] Bui, D., Zeng-Treitler, Q., 2014. Learning regular expressions for clinical text classification. Journal of the American Medical Informatics Association 21, 850–857. doi:10.1136/amiajnl-2013-002411.

[7] Casagrande, A., Gotti, F., Lapalme, G., 2017. Cerebra, un système de recommandation de candidats pour l'e-recrutement, in: AISR 2017, Paris. URL: http://rali.iro.umontreal.ca/rali/sites/default/files/publis/cerebra_systeme_recommandation_erecrutement.pdf.

[8] Castano, E., Verdaguer, I., Hilferty, J., 2019. Using metaphor to explore the organizational patterns of expository writing. Cuadernos de Investigación Filológica 46, 3–26. doi:10.18172/cif.3635.

[9] Chifu, E.S., Chifu, V.R., Popa, I., Salomie, I., 2017. A system for detecting professional skills from resumes written in natural language, in: 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, Cluj-Napoca, Romania. pp. 189–196. doi:10.1109/ICCP.2017.8117003.

[10] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. URL: https://aclanthology.org/N19-1423, doi:10.18653/v1/N19-1423.

[11] Dinarelli, M., Tellier, I., 2018. New recurrent neural network variants for sequence labeling, in: Gelbukh, A. (Ed.), Computational Linguistics and Intelligent Text Processing, Springer International Publishing, Cham. pp. 155–173. doi:10.1007/978-3-319-75477-2_10.

[12] Dixit, S., Verma, N., 2020. Intelligent condition based monitoring of rotary machines with few samples. IEEE Sensors Journal 20, 14337–14346. doi:10.1109/JSEN.2020.3008177.

[13] dos Santos, B.S., Steiner, M.T.A., Fenerich, A.T., Lima, R.H.P., 2019. Data mining and machine learning techniques applied to public health problems: A bibliometric analysis from 2009 to 2018. Computers and Industrial Engineering 138, 106120. doi:10.1016/j.cie.2019.106120.

[14] Dramé, K., Diallo, G., Delva, F., Dartigues, J.F., Mouillet, E., Salamon, R., Mougin, F., 2014. Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology: An application to alzheimer's disease. Journal of Biomedical Informatics 48, 171 – 182. doi:10.1016/j.jbi.2013.12.013.

[15] Franchina, L., Sergiani, F., 2020. High quality dataset for machine learning in the business intelligence domain, in: Bi, Y., Bhatia, R., Kapoor, S. (Eds.), Intelligent Systems and Applications, Springer International Publishing, Cham. pp. 391–401. doi:10.1007/978-3-030-29516-5_31.

[16] Govindarajan, V . S., Chen, B. T., Warholic, R., Erk, K., & Li, J.J. (2020). Help! Need Advice on Identifying Advice. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5295–5306. doi:10.48550/arXiv.2010.02494.

[17] Graves, A., 2008. Supervised sequence labelling with recurrent neural networks. Ph.D. thesis. Technical University Munich. URL: https://mediatum.ub.tum.de/doc/1289309/401810.pdf.

[18] Hakopov, Z., Mironov, D., Savic, D., Svetashova, Y., 2018. Automated kos-based subject indexing in INIS, in: Mayr, P., Tudhope, D., Busch, J.A., Golub, K., Hlava, M.M.K., Zeng, M. (Eds.), Proceedings of the 18th European Networked Knowledge Organization Systems (NKOS) Workshop co-located with the 22nd International Conference on Theory and Practice of Digital Libraries 2018 (TPDL 2018), Porto, Portugal, September 13, 2018, CEUR-WS.org. pp. 17–28. URL: http://ceur-ws.org/Vol-2200/paper2.pdf.

[19] Hou, L., Zhang, J., Wu, O., Yu, T., Wang, Z., Li, Z., Gao, J., Ye, Y., Yao, R., 2022. Method and dataset entity mining in scientific literature: A cnn + bilstm model with self-attention. Knowledge-Based Systems 235, 107621. URL: https://www.sciencedirect.com/science/article/pii/S0950705121008832, doi:10.1016/j.knosys.2021.107621.

[20] Inkeaw, P., Udomwong, P., Chaijaruwanich, J., 2021. Density based semi-automatic labeling on multi-feature representations for ground truth generation: Application to handwritten character recognition. Knowledge-Based Systems 220, 106953. URL: https://www.sciencedirect.com/science/article/pii/S0950705121002161, doi:10.1016/j.knosys.2021.106953.

[21] Ji, B., Liu, R., Li, S., Yu, J., Wu, Q., Tan, Y., Wu, J., 2019. A hybrid approach for named entity recognition in chinese electronic medical record. BMC Medical Informatics and Decision Making 19, 64. doi:10.1186/s12911-019-0767-2.

[22] Jumadinova, J., Bonham-carter, O., Zheng, H., Camara, M., Shi, D., 2020. A novel framework for biomedical text mining. Journal on Big Data 2, 145–155. doi:10.32604/jbd.2020.010090.

[23] Kang, Z., Catal, C., Tekinerdogan, B., 2020. Machine learning applications in production lines: A systematic literature review. Computers and Industrial Engineering 149, 106773. doi:10.1016/j.cie.2020.106773.

[24] Kato, T., Abe, K., Ouchi, H., Miyawaki, S., Suzuki, J., Inui, K., 2020. Embeddings of label components for sequence labeling: A case study of fine-grained named entity recognition, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Online. pp. 222–229. URL: https://aclanthology.org/2020.acl-srw.30.pdf.

[25] Kessler, R., Béchet, N., Roche, M., Torres-Moreno, J.M., El-Bèze, M., 2012. A hybrid approach to managing job offers and candidates. Information Processing and Management 48, 1124 – 1135. doi:10.1016/j.ipm.2012.03.002.

[26] Lavergne, T., Cappé, O., Yvon, F., 2010. Practical very large scale CRFs, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden. pp. 504–513. URL: https://aclanthology.org/P10-1052.

[27] Li, C., Bao, Z., Li, L., Zhao, Z., 2020a. Exploring temporal representations by leveraging attention-based bidirectional LSTM-RNNs for multi-modal emotion recognition. Information Processing and Management 57. doi:10.1016/j.ipm.2019.102185.

[28] Li, Z., Yang, Z., Xiang, Y., Luo, L., Sun, Y., Lin, H., 2020b. Exploiting sequence labeling framework to extract document-level relations from biomedical texts. BMC Bioinformatics 21. doi:10.1186/s12859-020-3457-2.

[29] Lin, J.C.W., Shao, Y., Djenouri, Y., Yun, U., 2021. Asrnn: A recurrent neural network with an attention model for sequence labeling. Knowledge-Based Systems 212, 106548. doi:10.1016/j.knosys.2020.106548.

[30] Lin, J.C.W., Shao, Y., Zhang, J., Yun, U., 2020. Enhanced sequence labeling based on latent variable conditional random fields. Neurocomputing 403, 431 – 440. doi:10.1016/j.neucom.2020.04.102.

[31] Ma, X., Hovy, E., 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany. pp. 1064–1074. doi:10.18653/v1/P16-1101.

[32] McCue, C., 2015. Chapter 12 - risk and threat assessment, in: McCue, C. (Ed.), Data Mining and Predictive Analysis (Second Edition). second edition ed.. Butterworth-Heinemann, Boston, pp. 257 – 282. doi:10.1016/B978-0-12-800229-2.00012-2.

[33] Meyer, B.J., Rice, G.E., 1982. The interaction of reader strategies and the organization of text. Text - Interdisciplinary Journal for the Study of Discourse 2, 155–192. doi:10.1515/text.1.1982.2.1-3.155.

[34] Mitrofan, M., Barbu Mititelu, V., Mitrofan, G., 2018. Towards the construction of a gold standard biomedical corpus for the romanian language. Data 3, 53. doi:10.3390/data3040053.

[35] Mittal, V., Mehta, P., Relan, D., Gabrani, G., 2020. Methodology for resume parsing and job domain prediction. Journal of Statistics and Management Systems 23, 1265–1274. doi:10.1080/09720510.2020.1799583.

[36] Modaresnezhad, M., Iyer, L., Palvia, P., Taras, V., 2020. Information technology (it) enabled crowdsourcing: A conceptual framework. Information Processing and Management 57, 102135. doi:10.1016/j.ipm.2019.102135.

[37] Natalia, F., Cheria, D., Surya, S., 2020. An ontology-based approach to diagnosis and classification for an expert system in health and food, in: Maseleno, A., Othman, M. (Eds.), Ontological Analyses in Science, Technology and Informatics. IntechOpen, Rijeka. chapter 4. doi:10.5772/intechopen.88180.

[38] Nentidis, A., Krithara, A., Tsoumakas, G., Paliouras, G., 2020. Beyond mesh: Fine-grained semantic indexing of biomedical literature based on weak supervision. Information Processing and Management 57, 102282. doi:10.1016/j.ipm.2020.102282.

[39] Ramponi, A., van der Goot, R., Lombardo, R., Plank, B., 2020. Biomedical event extraction as sequence labeling, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online. pp. 5357–5367. doi:10.18653/v1/2020.emnlp-main.431.

[40] Sahrawat, D., Mahata, D., Zhang, H., Kulkarni, M., Sharma, A., Gosangi, R., Stent, A., Kumar, Y., Shah, R.R., Zimmermann, R., 2020. Keyphrase extraction as sequence labeling using contextualized embeddings, in: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham. pp. 328–335. doi:10.1007/978-3-030-45442-5_41.

[41] Selway, M., Grossmann, G., Mayer, W., Stumptner, M., 2013. Formalising natural language specifications using a cognitive linguistics/configuration based approach, in: 2013 17th IEEE International Enterprise Distributed Object Computing Conference, pp. 59–68. doi:10.1109/EDOC.2013.16.

[42] Shemshadsara, Z.G., Ahour, T., Tamjid, N.H., 2019. Raising text structure awareness: A strategy of improving efl undergraduate students' reading comprehension ability. Cogent Education 6, 1644704. doi:10.1080/2331186X.2019.1644704.

[43] Sulova, S., Todoranova, L., Penchev, B., Nacheva, R., 2017. Using text mining to classify research papers, in: 17th International Multidisciplinary Scientific GeoConference SGEM 2017, STEF92 Technology. pp. 647–654. doi:10.5593/sgem2017/21/S07.083.

[44] Sychev, O.A., Penskoy, N.A., 2019. Method of lemmatizer selections in multiplexing lemmatization. IOP Conference Series: Materials Science and Engineering 483, 012091. doi:10.1088/1757-899x/483/1/012091.

[45] Taktek, E., Thakker, D., 2020. Pentagonal scheme for dynamic xml prefix labelling. Knowledge-Based Systems 209, 106446. doi:10.1016/j.knosys.2020.106446.

[46] Vallet, D., Fernández, M., Castells, P., 2005. An ontology-based information retrieval model, in: Gómez-Pérez, A., Euzenat, J. (Eds.), The Semantic Web: Research and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 455–470. doi:10.1007/11431053_31.

Appendix A. Example of the input for job dissemination

Web Developer (HTML/CSS) H/F
Xtramile company is looking for a passionate web developer.
You will be working with various technologies including Craft CMS and Shopify to bring our new website and user experience to life.
About you: Education: Master degree.
Experience with front-end development technologies including: Experience building responsive front ends that are usable across various devices/screen sizes. Comfortable with mobile first design. Experience with Javascript animation libraries. Familiarity with Unity. You are autonomous and manage the team work. You have this skills: HTML5, CSS3/SCSS/BEM, JavaScript, GIT, Craft CMS (bonus not essential)

(a) Job offer plain text

<?xml version="1.0" encoding="UTF-8" ?>

<root><Occupation> Web Developer </Occupation> <Occupation> web developer </Occupation> <Education> Master degree </ Education > <Experience> front-end development technologies </ Experience > < Experience > building responsive front ends that are usable across various devices/screen sizes </ Experience > < Experience > Javascript animation libraries </ Experience > < Main_tasks > working with various technologies including Craft CMS and Shopify to bring our new website and user experience to

life </ Main_tasks > < Hard_skills > Unity </ Hard_skills > < Hard_skills > HTML5 </ Hard_skills > < Hard_skills > CSS3 </ Hard_skills > < Hard_skills > SCSS </ Hard_skills > < Hard_skills > BEM </ Hard_skills > < Hard_skills > JavaScript </ Hard_skills > < Hard_skills > GIT </ Hard_skills > < Hard_skills > Craft CMS </ Hard_skills > < Soft_skills > autonomous </ Soft_skills > < Soft_skills > team work </ Soft_skills > </root>

(**b**) XML data

Fig. A.1. From a job offer plain text to XML data for dissemination on the job boards.

Appendix B. Rules

Notations:

l: the name of a label

seq: a sequence from the plain text

label(seq): label of the sequence *seq*. *label* and *sequences* are inverse functions

next(seq): sequence that directly follows the sequence *seq*

prev(seq): sequence that directly precedes the sequence *seq*

concat(seq1, seq2): function that returns the concatenation of both sequences *seq1* and *seq2*

all_labels: list of all the labels

'O': label that represents Other, i.e., no annotation

Lists of terms:

connector_list = ['or', 'and', 'without', 'with']

date_list = ['as soon as possible', 'soon']

pay_list = ['gross, 'net']

period_list = ['month', 'year', 'monthly', 'day', 'daily']

punctuation_list = [',', ')', '(', '.', ';', '\n', '\t', '\n\n']

salary_periodicity_list = ['month', 'year', 'monthly', 'day', 'daily']

salary_list = ['attractive', 'negotiable']

status_list = ['mandatory', 'optional', 'required']

unit_list = ['dollar', 'dollars', 'euro', 'euros', 'eur', '€', '$']

Table B.1

Domain-specific rules for manual annotation.

Category of rules	Concerned labels	The rules	Logical statement
Domain-specific rules	occupation	If a sequence is 'H/F', the label of the previous sequence is the occupation.	**IF** *seq* = 'H/F' **THEN** *label(prev(seq))*= 'occupation' **AND** *label('seq)='O'*
Domain-specific rules	experience	If any of the words 'mandatory' or 'optional' appear after an 'experience' label, then the complete sequence is associated to the label 'experience'.	**IF** *seq* ϵ status_list **AND** label(*prev(seq)*) = 'experience' **THEN** *label(concat(prev(seq),seq))* = 'experience'

Domain-specific rules	education	If any of the words 'mandatory' or 'optional' appear after an 'education' label, then the complete sequence is associated to the label 'education'.	**IF** *seq* ϵ status_list **AND** label(*prev(seq)*) = 'education' **THEN** *label(concat(prev(seq),seq))* = 'education'

Table B.2
General rules for manual annotation.

Category of rules	Concerned labels	The rules	Logical statement
Natural language rules	date start, date end, schedule	If any of the words present in date_list appear after a 'date_start' label, then the complete sequence is associated to the label 'date_start'.	**IF** *seq* ϵ date_list **AND** *label(prev(seq))* = 'date_start' **THEN** *label(concat(prev(seq),seq))* = 'date_start'
Natural language rules	contract duration	If any of the words present in period_list appear after a 'contract_duration' label, then the complete sequence is associated to the label 'contract_duration'.	**IF** seq ϵ period_list **AND** label(*prev(seq)*) = 'contract_duration' **THEN** *label(concat(prev(seq),seq))* = 'contract_duration'
Natural language rules	experience duration	If any of the words present in period_list appear after an 'experience_duration' label, then the complete sequence is associated to the label 'experience_duration'.	**IF** seq ϵ period_list **AND** label(*prev(seq)*) = 'experience_duration' **THEN** *label(concat(prev(seq),seq))* = 'experience_duration'
Natural language rules	salary	If any of the words present in unit_list appear after a 'salary' label, then the complete sequence is associated to the label 'salary'.	**IF** seq ϵ unit_list **AND** *label(prev(seq))* = 'salary' **THEN** *label(concat(prev(seq),seq))* = 'salary'
Natural language rules	salary	If any of the words present in salary_list appear after a 'salary' label, then the complete sequence is associated to the label 'salary'.	**IF** *seq* ϵ salary_periodicity_list **THEN** *label(seq)* = 'O'
Natural language rules	salary	If any of the words 'attractive' or 'negotiable' appear after a 'salary' label, then the complete sequence is associated to the label 'salary'.	**IF** seq ϵ salary_list **AND** label(*prev(seq)*) = 'salary' **THEN** *label(concat(prev(seq),seq))* = 'salary'
Natural language rules	salary	If any of the words 'gross' or 'net' appear after a 'salary' label, then the complete sequence is associated to the label 'salary'.	**IF** *seq* ϵ pay_list **AND** label(*prev(seq)*) = 'salary' **THEN** *label(concat(prev(seq),seq))* = 'salary'
Punctuation rules	All the labels	Consider in the sequence annotation the line skipped to keep the context, the punctuation marks, etc.	**IF** *seq* ϵ punctuation_list **AND** *label(prev(seq))* = *label(next(seq))* **THEN** *label(seq)* = *label(prev(seq))*
Sentence connectors	All the labels	If there is a sentence connector in a sequence that represents the same label,	IF *seq* ϵ punctuation_list **AND** *label(prev(seq))* = *label(next(seq))*

		consider it in the annotation.	**THEN** *label(seq) = label(prev(seq))*

Appendix C. Set of label distribution

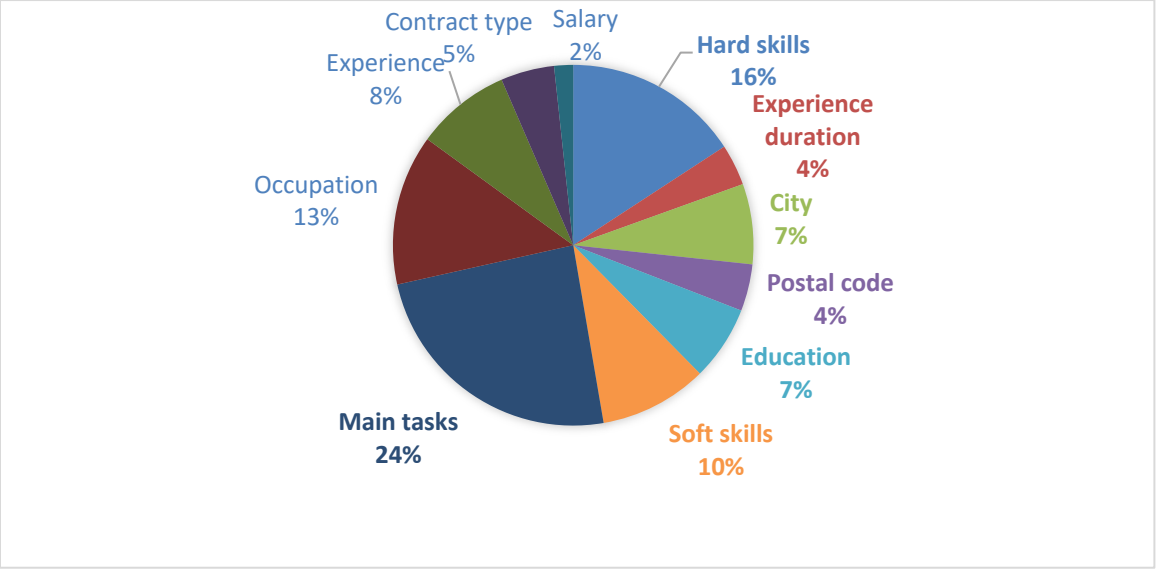

Fig. C.1. Distribution of the labels in job offers.

Appendix D. Example of an annotated offer in the tool Dataturks

Fig. D.1. Example of a job offer manually annotated using the Dataturks software.