Identifying Repeated Sections within Documents

Girish K. Palshikar, Sachin Pawar, Rajiv Srivastava, and Mahek Shah

gk.palshikar@tcs.com, sachin7.p@tcs.com,
rajiv.srivastava@tcs.com, shah.mahek@tcs.com
TCS Research & Innovation, Pune, India - 411013

Abstract. Identifying sections containing a logically coherent text about a particular aspect is important for fine-grained IR, question-answering and information extraction. We propose a novel problem of identifying repeated sections, such as project details in resumes and different sports events in the transcript of a news broadcast. We focus on resumes and present four techniques (2 unsupervised, 2 supervised) for automatically identifying repeated project sections. The knowledge-based method is modeled after the human way closely. The other methods are based on integer linear programming and sequence labeling. The proposed techniques are general and can be used for identifying other kinds of repeated sections (and even non-repeating sections) in different types of documents. We compared the four methods on a dataset of resumes of IT professionals and also evaluated the benefits of identifying such repeated sections in practical IR tasks. To the best of our knowledge, this paper is the first to propose and solve the problem of repeated sections identification.

Keywords: Section identification · Fine-grained IR · Resume searching.

1 Introduction

Many semi-structured documents (e.g., resumes, news, medical reports, court judgments, financial analyst reports) are loosely organized in the form of *sections*, where each section contains logically coherent text about a particular aspect. E.g., a resume in the IT domain may contain sections related to Education, Employment History, Project Details, Trainings, Personal Information etc. Automatically identifying sections containing information of a specific type within particular types of documents is important for fine-grained IR, question-answering and information extraction. E.g., if we are looking for people having hands-on experience in "SQL Server" in at least 2 projects, we should look only into the Project Details section (where the candidate lists all different projects she has worked on), and not, say, into the Trainings section of the resume.

In a semi-structured document, a section is often identified with a unique *section number*, and a *section title*. However, this identification is not uniform, even in the same type of documents. For example, in a resume, the section corresponding to past work experience has many different titles, such as `Work History`, `Employment History`, `Work Experience`, `Experience Summary`, `Previous Employments` etc. Thus identifying the section on past work experience in a resume can become difficult. In this paper, we focus on another kind of section identification: "identifying repeated

Table 1: An excerpt from an example resume where multiple repeated project sections are marked. Actual organization names in the resume are masked for anonymity.

`...`	
`PROJECTS`	
`SSSS FFFFF (Cargill, Brazil), May 2018 Present`	Project1 begins
`Internal application for client for its sales team.`	Project1 continues
`Application will be used to create and submit orders and view order history for its customers.`	Project1 continues
`Currently in planning phase. I am responsible as technical lead of team and currently understanding the requirements.`	Project1 continues
`WWW GGGG KKKKK (Belgium), Apr 2017 Apr 2018`	Project2 begins
`Hospital app for nurses visiting patients at home and keep track of medical records.`	Project2 continues
`Successfully delivered the project in requisite time and currently in UAT for final release.`	Project2 continues
`I was responsible for leading the offshore team on technical and design level and as supervisor.`	Project2 continues
`Also responsible for application development and issue resolution support.`	Project2 continues
`DDDD (DDDD JJJ), Jan 2017 Mar 2017`	Project3 begins
`A job hunting application for job seekers.`	Project3 continues
`Using the app, users can register themselves as potential job seekers.`	Project3 continues
`They can search and apply for jobs, check and update profile and keep a track of their job applications.`	Project3 continues
`I was responsible for design, requirement gathering and application development.`	Project3 continues
`...`	

sections". In some type of documents, a same type of sections occurs multiple times. E.g., the resume of an IT person often has details on multiple projects that she has worked on, i.e., the resume contains multiple *project sections*, each of which gives the details of a particular project (see example in Table 1). All the project sections within a resume share common information elements, such as project title, client, duration, description, technology platforms used, role performed, team size etc. Different resumes may omit some of these information elements, and the order of these information elements may vary across resumes. These variations may even be present across project sections within the same resume.

As another example, legal contracts often contain many governance processes that describe the sequences of actions to be taken when certain triggers occur. Table 2 shows examples of two governance processes in a construction contract[1], which occur several pages apart and which are not explicitly marked as governance processes. Since contracts are complex and often hundreds of pages long, identifying such governance processes is critical for ensuring compliance to the contract. Treating each governance process as a "section", we can apply the methods in this paper to automatically identify them. Similar repeating sections occur in many other types of documents such as court judgements, analyst reports, annual reports, and news.

While there is some research about identifying different types of sections in a document, we could not find any work that identifies *repeated* (sub-)sections of the same

[1] `http://www.basnettdbr.com/pdfs/ConstCont_101117.pdf`

Table 2: Example of governance processes in a construction contract (GP1: Governance Process 1, GP2: Governance Process 2)

...	
Contractor shall give written notice as necessary to Owner and/or Bank that Contractor's Work is completed and, if required, shall supply lien releases or receipts evidencing payment in full be filed relative to Contractor's Work.	GP1 begins
Owner and/or Bank shall have the right to make final inspection of Contractor's work within seven days after receipt of notice of completion and upon acceptance thereof by Owner and Bank, payment shall be made of the remaining balance due.	GP1 continues
...	
In the event that required work cannot be priced in advance of completion of such work, an Additional Work Authorization shall be executed.	GP2 begins
Such orders shall describe work to be completed, and shall specify method of calculating additional fees, materials, labor and services to be charged upon completion, and become part of this contract.	GP2 continues
Payment shall be due upon presentation of Contractor invoice.	GP2 continues
Additional time required shall be estimated and stated within the Additional Work Authorization.	GP2 continues
...	

type. Hirohata et al. [4] proposed a CRF based approach to categorize sentences in scientific abstracts into 4 sections: objective, methods, results, and conclusions. Li et al. [8] is a supervised sequence labeling based approach for section classification which uses Hidden Markov Models. Shah et al. [10] proposes a CRF-based model for section identification, similar to ours. [12] propose a hierarchical information extraction framework to identify and label sections in a resume. [11] extracts different types of entities from resumes and uses them to improve ranking of resumes to match a job description; it does not deal with the problem of section identification. [2] is an unsupervised approach where they identify section labels using semantic relatedness (using word embeddings) between section title and contents and with predefined section labels. Guo et al. [3] proposed an unsupervised model which uses topic models to identify latent topics and their key linguistic features in input documents. Constraints are then induced from this information and sentences are mapped to their dominant section categories through a constrained unsupervised model. Most of these previous approaches are dependent on dominant topics or lexical contents of the sections. As our work focuses on identifying repeated sections of the same type, our approaches also consider the structure of individual sections in the form of occurrence patterns of various section markers.

In this paper, we present four techniques for automatically identifying such repeated project sections in resumes: (i) knowledge-based, (ii) integer linear programming based (both *unsupervised*), (iii) sequence labeling using CRF, and (iv) sequence labeling using LSTM (both *supervised*).

2 Problem Definition

A given resume is a sequence of N sentences $\langle 1, 2, \ldots N \rangle$. The task is to identify all the project sections in this given resume, in terms of the start and end sentence numbers for each. To simplify the problem, we assume that the project sections are contiguous and non-overlapping. Then it is enough to identify the start sentence number for each project section. For example, if i-th project section starts on say line 68, then the $(i-1)$-th project ends on sentence number 67. For the last project section, we use a simple heuristic rule to identify the end sentence number.

Let M denote the set of K *markers*. Each marker indicates the presence of a specific type of text. For the task of identifying project sections, we used $K = 9$ markers: $blankline, project, projectnum, client, duration, role, teamsize, description, technology$. Each of these markers detects the presence of some particular piece of information likely to be present in a project section. For example, the marker $project$ corresponds to keywords indicating the presence of the title or name of project; e.g., `Project, Module, Title, Name, Profile, Initiative`. We have defined a regular expression pattern to check whether or not a project marker is present in the given line. This regex not only detects the presence of required keywords but also performs additional checks; e.g., since `title` is also used for job designations, the regex checks for *absence* of designation indicating keywords (e.g., `manager, consultant`) near these keywords. Thus we have K Boolean arrays each of length N; for example, the array $project[17] = 1$ implies that a project marker is present on sentence 17 of the given resume document. A single sentence may contain 0, 1 or more than one markers.

3 Repeated Sections Identification

3.1 Knowledge-based Approach

A human HR executive can easily spot the project sections in any given resume. Given the endless variations in which the projects are written, this knowledge is quite non-trivial and does not consist of simple rules. In fact, we found that the humans have a dynamic, context-sensitive way of *rearranging* the project sections boundaries. We have tried to capture this expert knowledge in the form of an algorithm (Algorithm 1). The basic idea is that the human reader identifies the markers (already discussed), and then rearranges the project section starting point by understanding the spatial relationships among the marker positions. The essence of this rearrangement is as follows. It is usually true that a project section begins with the $project$ marker (e.g., project title), and all other markers (e.g., $client$, $teamsize$ etc.) come afterwards within the project section. But occasionally some of the markers may occur just before the project title. Thus we need to recognize such deviations from the typical sequence of markers within a project section, and adjust the starting sentence of the project section accordingly. Since a lot of such variations among marker sequences occur in practical resumes, there is much heuristic post-processing still required after algorithm $identify\_project\_sections$ (Algorithm 1), which we have omitted.

3.2 ILP based Approach

We model the "repeated section identification" problem using the Integer Linear Programming (ILP) formulation. Table 3 depicts the input parameters, the constraints and the objective used for the ILP formulation. The 9 boolean arrays corresponding to

```
input  : $S = \{s_1, \ldots, s_N\}$ sentences in given resume
input  : $project, projectnum, client, duration, role, teamsize,$
         $description, technology$ : Boolean arrays of length $N$
output: $P$ Boolean array of length $N$, $P[i] = 1$ if some project section starts at line $i$
$C$ := array of $N$ tuples, initialized with (0,NULL)
for $i = 1$ to $N$ do
    if $project[i] \wedge \exists$ another marker in $i \pm K$ then  $C[i] = (1, project)$
for $i = 1$ to $N$ do
    if $client[i] \wedge C[j] \neq 1$ for $i - K \leq j \leq i \wedge \exists$ another marker in $i \pm K$ then
        $C[i] = (1, client)$
for $i = 1$ to $N$ do
    if $duration[i] \wedge C[j] \neq 1$ for $i - K \leq j \leq i \wedge \exists$ another marker in $i \pm K$ then
        $C[i] = (1, duration)$
for $i = 1$ to $N$ do
    if $C[i] == (1, project)$ then
        Sequentially examine previous 10 sentences starting at $j = i - 1$ and stopping at
        any $j$ where $j$-th sentence is either blank or does not contain any marker nor any
        ':';
        if $j == i - 11 \wedge isblank(j)$ then $P[j] = 1$
        else if $j == i - 1 \wedge \exists i - 4 \leq k \leq i - 2$ s.t. $project[k] \wedge$ all sentences from
        $k + 1$ to $i - 1$ are blank then $P[k] = 1$
        else if $j < i \wedge j \geq i - 10$ then $P[j + 1] = 1$
        else $P[i] = 1$
    else if $C[i] == (1, client) \vee C[i] == (1, duration)$ then
        Sequentially examine previous 10 sentences starting at $j = i - 1$ and stopping at
        any $j$ where both $j$, $(j - 1)$-th sentences are blank or $j$-th sentence contains any
        marker;
        if $j < i \wedge j \geq i - 10$ then $P[j + 1] = 1$
        else if $j == i - 11 \wedge \exists j < k < i$ s.t. isblank(k) then $P[k + 1] = 1$
        else if $j == i - 11 \wedge isblank(j)$ then $P[j] = 1$
for $i = 1$ to $N$ do
    if $P[i] == 1 \wedge$ this section marked due to $project$ marker $\wedge (i - 1)$-th sentence has
    $< 8$ words $\wedge$ isblank($i - 2$) then
        $P[i - 1] = 1, P[i] = 0$
```

Algorithm 1: $identify\_project\_sections$

project section markers described earlier are the input parameters. Another input for the ILP formulation is S, the maximum number of project sections possible for the current resume. The **output representation** is in the form of two matrices (x and y) of $N \times S$ binary variables. If j^{th} project section begins at the k^{th} sentence, then the j^{th} column of x will contain all 0's *before* the k^{th} row and all 1's *after* that till the end. Similarly, if j^{th} project section ends at the k^{th} sentence, then the j^{th} column of y will contain all 0's before the k^{th} row and all 1's after that till the end. In other words, x is used to mark the beginning of individual project sections and y is used to mark the end. Hence, $(x[i, j] - y[i, j])$ will be 1 if and only if i^{th} sentence is part of j^{th} section. Also, by definition of x and y, $(x[i, j] - y[i, j])$ will be 1 for consecutive sentences only. The

objective is to minimize following 3 terms:
•T_1 = Number of project markers which are not part of any section: $\sum_{j=1}^{S}(x[i,j] - y[i,j])$ would be 0 only for sentences (i's) which are not part of any section. Here, $duration$ and $technology$ markers are not considered as they also occur outside project sections (e.g., $duration$ marker may be present in employment history).
•T_2 = Number of project sections. $x[N,j]$ is 1 if and only if j^{th} section is identified. Hence, the term $\sum_{j=1}^{S} x[N,j]$ simply counts the number of project sections identified.
•T_3 = Sentence numbers in which $project$, $projectnum$ and $duration$ markers occur in each section, relation to the sentence number corresponding to the beginning of that section. The term $(x[i,j] - x[i-1,j])$ will be 1 for one and only one i if j^{th} section is present, and hence the term $\sum_{i=2}^{N} i \cdot (x[i,j] - x[i-1,j]))$ corresponds to the first sentence of the j^{th} section.

Various **constraints** are defined to capture different desired properties of the output project sections. Table 3 describes these constraints in detail. A separate ILP program is created for each resume and is solved to compute optimal feasible values for output variables, which in turn translate to predicted project sections.

3.3 Sequence Labeling Approaches

In addition to the unsupervised approaches (knowledge-based and ILP-based), we also explored two supervised approaches. Here, we model the "repeated section identification" problem as a sequence labeling task where an appropriate label is assigned to each element in a sequence. Each resume is represented as a sequence of sentences and 3 labels (B, I and O) are used to represent the project section information. First sentence in each project section is labeled with B and all subsequent sentences within the section are labeled with I. Sentences which are not part of any project section are labeled with O. In order to learn a sequence labelling model, we explore following two approaches:

Conditional Random Fields (CRF): In this approach, each sentence is represented by a set of features which are designed to capture various characteristics of the sentence. Some of the representative features are as follows: presence of the project section markers in current, previous and next sentences, number of consecutive blank lines before and after each sentence, number of words in current sentence, distinct words present in current sentence. We use Conditional Random Fields (CRF) [6] for training a sequence labeling model, which is used to predict BIO labels for any unseen resume and predicted project sections can be derived from these labels.

Long Short-Term Memory (LSTM): The CRF-based approach requires explicit feature engineering for the task. Also, the markers-based features are dependent on accuracy and coverage of the regular expressions used to identify the markers. Hence, we also developed an LSTM-based [5] approach which bypasses the need for explicit feature engineering. Here, the model for section identification is built in two phases. Initially, our aim is to learn embedded representations for sentences in resumes. For this purpose, we train an LSTM-based sequence autoencoder [1] which consists of an

Table 3: ILP Formulation

Input Parameters:
N : No. of sentences in the resume, S : Max. no. of project sections possible, $project$, $client$, $projectnum$, $duration$, $role$, $teamsize$ and $description, technology$: arrays of length N
Decision Variables:
x : Binary matrix of size $N \times S$, $x[i,j] = 1, \forall i \geq k$ s.t. j^{th} section begins at the k^{th} sentence y : Binary matrix of size $N \times S$, $y[i,j] = 1, \forall i \geq k$ s.t. j^{th} section ends at the k^{th} sentence
Minimize: $T_1 + T_2 + T_3$
$T_1 = \sum_{i=1}^{N}((project[i]+client[i]+role[i]+teamsize[i]+description[i]+projectnum[i]) \cdot (1-\sum_{j=1}^{S}(x[i,j]-y[i,j]))); T_2 = \sum_{j=1}^{S} x[N,j]; T_3 = 0.001 \cdot \sum_{j=1}^{S}(\sum_{i=1}^{N} i \cdot (project[i] + projectnum[i] + client[i] + duration[i]) \cdot (x[i,j]-y[i,j]) - \sum_{i=2}^{N} i \cdot (x[i,j]-x[i-1,j]))$
Constraints:
//C_0 to C_3: Ensure sanity of the output $C_0 : x[i-1,j] \leq x[i,j], \forall(i,j), 2 < i < N, 1 < j < S$ $C_1 : y[i-1,j] \leq y[i,j], \forall(i,j), 2 < i < N, 1 < j < S$ $C_2 : x[i,j] \geq y[i,j], \forall(i,j), 1 < i < N, 1 < j < S$ $C_3 : y[i,j-1] \geq x[i,j], \forall(i,j), 1 < i < N, 1 < j < S$ //C_4: Ensures that each individual section contains at least two markers of any type $C_4 : \sum_{i=1}^{N}(project[i] + client[i] + role[i] + teamsize[i] + description[i] + duration[i] + projectnum[i] + technology[i]) * (x[i,j] - y[i,j])) \geq 2 \cdot x[N,j], \forall j, 1 < j < S$ //C_5: Ensures that each individual section contains at least one marker of $project, client, duration, projectnum$ $C_5 : \sum_{i=1}^{N}(project[i] + client[i] + duration[i] + projectnum[i]) * (x[i,j] - y[i,j])) \geq x[N,j], \forall j, 1 < j < S$ //C_6 to C_{12}: Ensure that each individual section does not contain repeated markers of certain types $C_6 : \sum_{i=1}^{N} client[i] \cdot (x[i,j] - y[i,j]) \leq 1, \forall j, 1 < j < S$ $C_7 : \sum_{i=1}^{N} duration[i] \cdot (x[i,j] - y[i,j]) \leq 1, \forall j, 1 < j < S$ $C_8 : \sum_{i=1}^{N} teamsize[i] \cdot (x[i,j] - y[i,j]) \leq 1, \forall j, 1 < j < S$ $C_9 : \sum_{i=1}^{N} projectnum[i] \cdot (x[i,j] - y[i,j]) \leq 1, \forall j, 1 < j < S$ $C_{10} : \sum_{i=1}^{N} description[i] \cdot (x[i,j] - y[i,j]) \leq 1, \forall j, 1 < j < S$ $C_{11} : \sum_{i=1}^{N} project[i] \cdot (x[i,j] - y[i,j]) \leq 2, \forall j, 1 < j < S$ $C_{12} : \sum_{i=1}^{N} role[i] \cdot (x[i,j] - y[i,j]) \leq 2, \forall j, 1 < j < S$ //C_{13}: Ensures that a project section follows a $blankline$ or begins with $project, projectnum$ or $duration$ markers $C_{13} : (x[i,j] - x[i-1,j]) \leq (1 - (1 - blankline[i-1]) * (1 - projectnum[i]) * (1 - project[i]) * (1 - duration[i])), \forall(i,j), 2 < i < N, 1 < j < S$

LSTM encoder layer and another LSTM decoder layer. The detailed architecture is depicted in the Figure 1. Here, a sequence of words in a sentence is passed through the encoder LSTM layer, so that the output of the final time step provides the representation of the whole sentence. Words are represented using 100 dimensional pre-trained GloVe [9] word vectors. The decoder LSTM layer then tries to reconstruct the same sequence of the words using this representation. The model is trained in an unsupervised fashion to minimize the Mean Squared Error at the decoder output. We train this model on a large corpus of 810 resumes having more than 140,000 sentences, using 100-dim word vectors and 200-dim vectors for sentence representations.

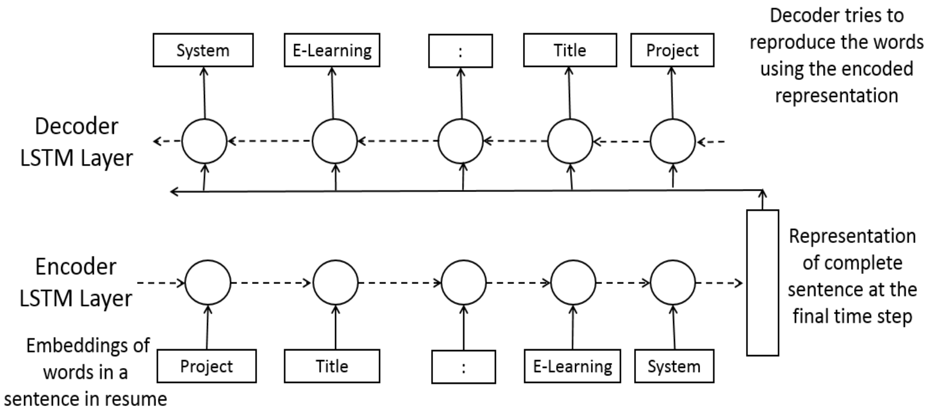
Decoder tries to reproduce the words using the encoded representation
System
E-Learning
:
Title
Project
Decoder LSTM Layer
Representation of complete sentence at the final time step
Encoder LSTM Layer
Embeddings of words in a sentence in resume
Project
Title
:
E-Learning
System

Fig. 1: LSTM-based sequence autoencoder used to learn embedded representations for sentences in resumes

Using the encoder LSTM in our sequence autoencoder, we can now get an embedded representation for any new or unseen resume sentence. As discussed earlier, a resume is a sequence of sentences and our aim is to assign an appropriate label (from BIO labels) to each sentence to identify project sections. For this purpose, we design another Bi-directional LSTM model [7] where input for the Bi-LSTM layer is a sequence of embedded representations of sentences in a resume. For the i^{th} sentence, Bi-LSTM provides two representation vectors: (i) *previous* context vector which captures the context from the first to $(i-1)^{th}$ sentences, and (ii) *next* context vector which captures the context from $(i+1)^{th}$ till the last sentence in the resume. These two context vectors are concatenated and passed to a softmax layer for final prediction of BIO labels. The detailed architecture of this model is depicted in Figure 2. This model is then trained in a supervised fashion (similar to CRF) using labelled dataset of resumes annotated with gold-standard project sections. Once the model is learned, it can be used for predicting BIO labels for any unseen resume and predicted project sections can be derived from these labels. Moreover, we tried another variant of this Bi-LSTM based model, where for each sentence, we augment the sentence representation (provided by the LSTM encoder layer in our sequence autoencoder) with a vector representing the presence of project section markers. This vector is a binary vector containing K bits corresponding to each project section marker. If any marker is present in a sentence, then its corresponding bits are set to 1, otherwise they are set to 0. The value of $K = 10$ was empirically found to be suitable.

4 Experimental Evaluation

In this section, we describe the dataset details as well as intrinsic and extrinsic evaluation strategies.

4.1 Datasets

We manually annotated 366 resumes for project section information. For each resume, we identified sentence numbers corresponding to the first sentences of project sections. The dataset was partitioned into two parts: D_1 (206 resumes) and D_2 (160 resumes). The resumes in the dataset D_1 were used to fine-tune the patterns for project markers as well as the rules used in knowledge-based approach. The supervised sequence labeling approaches (CRF and LSTM-based approaches) are trained on D_1. The dataset D_2 is a blind test set for all approaches.

4.2 Intrinsic Evaluation

For intrinsic evaluation of the proposed approaches, we compare the predicted project sections with the manually annotated (gold-standard) project sections at 3 different evaluation levels (see Table 4):

- **Strict**: If first sentence number of a predicted section matches that of a gold-standard section, a true positive (TP) is counted. Unmatched predicted sections are counted as false positives (FP); unmatched gold-standard sections are false negatives (FN).
- **Lenient1**: If first sentence number of a predicted section is within ± 3 sentences of that of a gold-standard section, a TP is counted. A predicted section is counted as FP only if there is no gold-standard section starting within ± 3 sentences. A gold-standard section is counted as FN only if there is no predicted section starting within ± 3 sentences.
- **Lenient2**: When the first sentence number of a predicted section is within ± 20 words of that of a gold-standard section, a TP is counted. FPs/FNs are counted analogously.

Table 4 shows the comparative performance of the proposed approaches. All the approaches are evaluated on both the datasets D_1 and D_2. For the supervised approaches

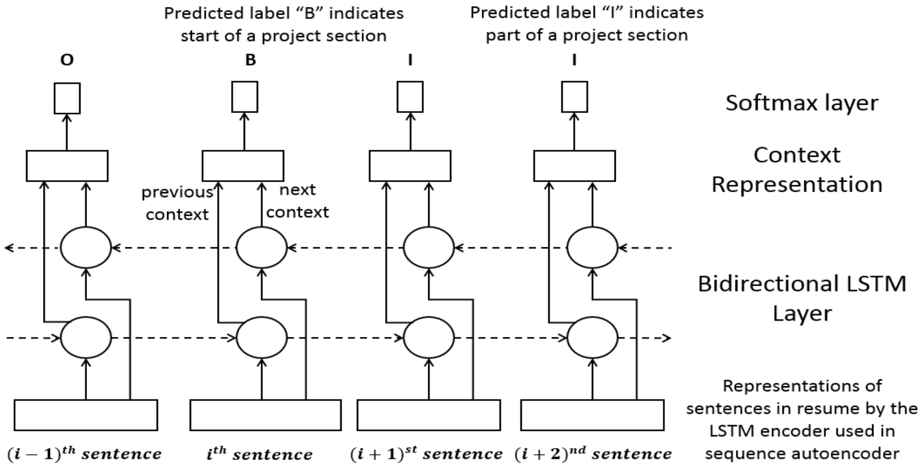
Predicted label "B" indicates start of a project section
Predicted label "I" indicates part of a project section
O
B
I
I
Softmax layer
Context Representation
previous context
next context
Bidirectional LSTM Layer
Representations of sentences in resume by the LSTM encoder used in sequence autoencoder
$(i-1)^{th}$ sentence
i^{th} sentence
$(i+1)^{st}$ sentence
$(i+2)^{nd}$ sentence

Fig. 2: Bi-LSTM sequence labelling model for assigning BIO labels to resume sentences

Table 4: Project section identification performance (Strict and Lenient Evaluations)

	Knowledge			**ILP**			**CRF**			**LSTM**			**LSTM**		
	based									w/o markers			with markers		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Strict-D_1	88.0	84.0	**86.0**	50.0	46.3	48.1	78.5	66.0	71.7	57.9	80.3	67.3	65.0	80.5	71.9
Lenient1-D_1	94.3	89.2	**91.7**	77.3	70.9	74.0	89.6	69.4	78.3	79.4	91.2	84.9	87.4	91.3	89.3
Lenient2-D_1	92.3	87.5	**89.8**	74.6	69.1	71.7	89.0	68.9	77.6	77.5	89.0	82.9	85.3	89.5	87.3
Strict-D_2	64.2	63.6	63.9	53.1	46.0	49.3	71.8	50.9	59.6	53.9	77.3	63.5	61.7	79.8	**69.6**
Lenient1-D_2	78.2	74.7	76.4	75.6	65.0	69.9	88.5	59.1	70.9	73.7	90.9	81.4	79.6	91.2	**85.0**
Lenient2-D_2	76.2	72.5	74.3	71.4	61.4	66.0	87.8	58.4	70.1	70.5	86.8	77.8	77.3	87.9	**82.2**

(CRF and LSTM-based), the dataset D_1 is used for training the models which are then applied on the dataset D_2. Also, these approaches are also evaluated on D_1 using 5-fold cross-validation. It can be observed that the knowledge-based approach outperforms all other approaches on D_1. But its performance degrades on D_2 which is the blind test set. Although, such performance degradation is observed for all approaches, it is more pronounced for the knowledge-based approach. As the rules in the knowledge-based approach are designed by observing project sections in D_1, this approach seems to have over-fitted on D_1. Although, ILP-based approach lags behind the knowledge-based approach, its performance is more consistent across D_1 and D_2. As ILP-based approach is a more principled way of representing the domain knowledge, it is also easier to maintain.

On the dataset D_2, the LSTM-based approach using markers outperforms all other approaches. Moreover, this approach provides more consistent performance across both the datasets D_1 and D_2. Although, the CRF-based approach lags behind in F1, it provides the best precision on D_2. In future, we plan to explore an ensemble of CRF and LSTM-based approaches to exploit the high-precision nature of CRF-based approach as well as the high-recall nature of LSTM-based approach.

The LSTM-based approach without using markers also outperforms knowledge-based, ILP-based and CRF-based approaches. It is important to note that this approach has the least dependence on the domain knowledge because it does not need information about project section markers. It only depends upon the sentence representations learned in an unsupervised manner using sequence autoencoder. Hence, the LSTM-based approach without markers can be easily re-trained for any other domain for identification of other types of repeated sections.

4.3 Extrinsic Evaluation

As mentioned earlier, several practical applications need project sections to be identified in resumes. So we used the resumes with or without identification of project sections to evaluate the output of some typical end-user queries. For example, a recruitment executive or project leader is interested in candidates who have used skill X in at least one project. Since the given skill X may occur outside project sections as well, one should only identify those candidates for whom X occurs within at least one project

Table 5: User queries with/without project sections.

Query ⇒	**SQL Server 2008**			**Hibernate**			**Web Sphere**		
Setting ⇓	P	R	F	P	R	F	P	R	F
S_1: No project sections	50.0	100.0	66.7	77.8	100.0	87.5	69.6	100.0	82.1
S_2: Knowledge-based	100.0	100.0	100.0	100.0	100.0	100.0	93.3	87.5	90.3
S_3: ILP	100.0	100.0	100.0	100.0	100.0	100.0	93.8	93.8	93.8
S_4: CRF	50.0	100.0	66.7	100.0	71.4	83.3	100.0	75.0	85.7
S_5: LSTM w/o markers	100.0	100.0	100.0	100.0	100.0	100.0	100.0	87.5	93.3
S_6: LSTM with markers	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.8	96.8

section. Not using project sections will retrieve all resumes in which X occurs *somewhere* in the resume, not necessarily within any project sections. The results are shown in Table 5 for 3 different skills. In setting S_1, the resumes were retrieved without using any project section information. For settings S_2 to S_6, the resumes were retrieved using the project section information predicted by knowledge-based approach, ILP-based approach, CRF-based approach, LSTM-based approach without markers and LSTM-based approach using markers, respectively. Using ground truth, we computed precision, recall and F-measure for these 6 settings. Other example queries where project sections are important are: *find candidates who have used skill X for at least 12 months* and *find candidates who have used skill X in role developer*.

5 Conclusions

In this paper, we proposed 2 unsupervised and 2 supervised methods for the novel problem of identifying repeated sections in a document (in particular, resumes). The proposed methods can also detect non-repeating sections containing specific types of information. Our knowledge-based method is interesting because it is modeled after the human ways of dealing with the same problem, but its drawback is that it is hard to maintain. The ILP based method is similar but more robust. We compared the four methods on a dataset of resumes of IT professionals. The 2 supervised methods based on CRF and LSTM also perform well, where the LSTM-based method outperforms all other methods on our blind test set. Though the CRF-based method underperforms the LSTM-based method, it achieves the highest precision among all the methods. In future, we plan to explore an ensemble of CRF and LSTM-based methods to exploit the high-precision nature of CRF-based method as well as the high-recall nature of LSTM-based method. We also evaluated the benefits of identifying such repeated sections in practical IR tasks. Topic-based section identification methods do not work well, because the same topics occur across different repeated sections. The problem proposed here is of wider interest for fine-grained IR and can be used to identify sections in a wide variety of documents, such as legal documents, news, financial reports, scientific papers and web pages.

References

1. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in neural information processing systems. pp. 3079–3087 (2015)
2. Garg, S., Singh, S.S., Mishra, A., Dey, K.: Cvbed: Structuring cvs using word embeddings. In: IJCNLP: Volume 2. vol. 2, pp. 349–354 (2017)
3. Guo, Y., Reichart, R., Korhonen, A.: Unsupervised declarative knowledge induction for constraint-based learning of information structure in scientific documents. TACL **3**(1), 131–143 (2015)
4. Hirohata, K., Okazaki, N., Ananiadou, S., Ishizuka, M.: Identifying sections in scientific abstracts using conditional random fields. In: IJCNLP:Volume-I (2008)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
6. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
7. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT. pp. 260–270 (2016)
8. Li, Y., Lipsky Gorman, S., Elhadad, N.: Section classification in clinical notes using supervised hidden markov model. In: Proc. 1st ACM Int. Health Informatics Symposium. pp. 744–750 (2010)
9. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
10. Shah, M., Palshikar, G., Srivastava, R.: New approaches of resume sectioning for automating talent acquisition. In: Fifth Int. Conf. Business Analytics and Intelligence (ICBAI) (2017)
11. Singh, A., Rose, C., Visweswariah, K., Chenthamarakshan, V., Kambhatla, N.: Prospect: a system for screening candidates for recruitment. In: CIKM. pp. 659–668 (2010)
12. Yu, K., Guan, G., Zhou, M.: Resume information extraction with cascaded hybrid model. In: ACL. pp. 499–506 (2005)