Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models

Changmao Li[♠], Elaine Fisher[♢], Rebecca Thomas[♡], Steve Pittard[♣], Vicki Hertzberg[♢], Jinho D. Choi[♠]

[♠]Department of Computer Science, Emory University, Altanta GA, USA
[♢]Nell Hodgson Woodruff School of Nursing, Emory University, Altanta GA, USA
[♣]Department of Biostatistics and Bioinformatics, Emory University, Altanta GA, USA
[♡]Georgia CTSA Clinical Research Centers, Emory Healthcare, Atlanta GA, USA

changmao.li@emory.edu, elaine.fisher@emory.edu, rebecca.s.thomas@emoryhealthcare.org
wsp@emory.edu, vhertzb@emory.edu, jinho.choi@emory.edu

Abstract

This paper presents a comprehensive study on resume classification to reduce the time and labor needed to screen an overwhelming number of applications significantly, while improving the selection of suitable candidates. A total of 6,492 resumes are extracted from 24,933 job applications for 252 positions designated into four levels of experience for Clinical Research Coordinators (CRC). Each resume is manually annotated to its most appropriate CRC position by experts through several rounds of triple annotation to establish guidelines. As a result, a high Kappa score of 61% is achieved for inter-annotator agreement. Given this dataset, novel transformer-based classification models are developed for two tasks: the first task takes a resume and classifies it to a CRC level (`T1`), and the second task takes both a resume and a job description to apply and predicts if the application is suited to the job (`T2`). Our best models using section encoding and multi-head attention decoding give results of 73.3% to `T1` and 79.2% to `T2`. Our analysis shows that the prediction errors are mostly made among adjacent CRC levels, which are hard for even experts to distinguish, implying the practical value of our models in real HR platforms.

1 Introduction

An ongoing challenge for Human Resource (HR) is the process used to screen and match applicants to a target job description with a goal of minimizing recruiting time while maximizing proper matches. The use of generic job descriptions not clearly stratified by the level of competence or skill sets often leads many candidates to apply every possible job, resulting in misuse of recruiter and applicant's time. A more challenging aspect is the evaluation of unstructured data such as resumes and CVs, which represents about 80% of the data processed daily, a task that is typically not an employer's priority given the manual effort involved (Stewart, 2019). The current practice for screening applications involves reviewing individual resumes via traditional approaches, that rely on string/regex matching. The scope of posted job positions varies by the hiring organization type, job level, focus area, and more. The latest advent in Natural Language Processing (NLP) enables the large-scale analysis of resumes (Deng et al., 2018; Myers, 2019). NLP models also allow for a comprehensive analyses on resumes and identification of latent concepts that may easily go unnoticed using a general manual process. This model's ability to infer core skills and qualifications from resumes can be used to normalize necessary content into standard concepts for matching with stated position requirements (Chifu et al., 2017; Valdez-Almada et al., 2018). However, the task of resume classification has been under-explored due to the lack of resources for individual research labs and the heterogeneous nature of job solicitations.

This paper presents new research that aims to help applicants identify the level of job(s) they are qualified for and to provide recruiters with a rapid way to filter and match the best applicants. For this study, resumes submitted to four levels of Clinical Research Coordinator (CRC) positions are used. To the best of our knowledge, this is the first time that resume classification is explored with levels of competence, not categories. The contributions of this work are summarized as follows:

- To create a high-quality dataset that comprises 3,425 resumes annotated with 5 levels of real CRC positions (Section 3).
- To present a novel transformer-based classification approach using section encoding and multi-head attention decoding (Section 4).
- To develop robust NLP models for the tasks of competence-level classification and resume-to-job_description matching (Section 5).

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8456–8466,
November 16–20, 2020.

Type	Description
CRC1	Manage administrative activities associated with the conduct of clinical trials. Maintain data pertaining to research projects, complete source documents/case report forms, and perform data entry. Assist with participant scheduling.
CRC2	Manage research project databases and development study related documents, and complete source documents and case report forms. Interface with research participants and study sponsors, determine eligibility, and consent study participants according to protocol.
CRC3	Independently manage key aspects of a large clinical trial or all aspects of one or more small trials or research projects. Train and provide guidance to less experienced staffs, interface with research participants, and resolve issues related to study protocols. Interact with study sponsors, monitor/report SAEs, and resolve study queries. Provide leadership in determining, recommending, and implementing improvements to policies and procedures.
CRC4	Function as a team lead to recruit, orient, and supervise research staff. Independently manage the most complex research administration activities associated with the conduct of clinical trials. Determine effective strategies for promoting/recruiting research participants and retaining participants in long term clinical trials.

Table 1: Descriptions (and general responsibilities) of the four-levels of CRC positions.

2 Related Work

Limited studies have been conducted on the task of resume classification. Zaroor et al. (2017) proposed a job-post and resume classification system that integrated knowledge base to match 2K resumes with 10K job posts. Sayfullina et al. (2017) presented a convolutional neural network (CNN) model to classify 90K job descriptions, 523 resume summaries, and 98 children's dream job descriptions into 27 job categories. Nasser et al. (2018) hierarchically segmented resumes into sub-domains, especially for technical positions, and developed a CNN model to classify 500 job descriptions and 2K resumes.

Prior studies in this area have focused on classifying resumes or job descriptions into occupational categories (e.g., data scientist, healthcare provider). However, no work has yet been found to distinguish resumes by levels of competence. Furthermore, we believe that our work is the first to analyze resumes together with job descriptions to determine whether or not the applicants are suitable for particular jobs, which can significantly reduce the intensive labor performed daily by HR recruiters.

3 Dataset

3.1 Data Collection

Between April 2018 and May 2019, the department of Human Resources (HR) at Emory University received about 25K applications including resumes in free text for 225 Clinical Research Coordinator (CRC) positions. A CRC is a clinical research professional whose role is integral to initiating and managing clinical research studies. There are four levels of CRC positions, CRC1–4, with CRC4 having the most expertise. Table 1 gives the descriptions about these four CRC levels.

Table 2 shows the statistics of the collected applications and the resumes. Out of the 24,933 applications, 89% are applied for the entry level positions, CRC1–2, that is expected since CRC3–4 positions require more qualifications (A). At any time, there are various positions posted for the same level from different divisions, cardiology, renal, infectious disease, etc. Thus, it is common to see resumes from the same applicant applying to several job postings within the same CRC level.

After removing duplicated resumes within the same level, 9,286 resumes remain, discarding 63% of the original applications (B). It is common to see the same applicant applying to positions across multiple levels. After removing duplicated resumes across all levels and retaining only the resumes to the highest level (e.g., if a person applied for both CRC1 and CRC2, retain the resume for only CRC2), 6,492 resumes are preserved, discarding additional 11% from the original applications (C).

	CRC1	CRC2	CRC3	CRC4	Total
A	13,794	8,415	2,238	486	24,933
B	4,779	3,005	1,106	396	9,286
C	2,961	2,250	885	396	6,492
B_r	2,730	1,702	696	234	5,362
C_r	1,477	1,172	542	234	3,425

Table 2: The counts of applications (A), unique resumes for each level (B), unique resumes across all levels (C), and resumes from B and C selected for our research while preserving level proportions (B_r and C_r).

For our research, we carefully select 3,425 resumes from C by discarding ones that are not clearly structured (e.g., no section titles) or contain too many characters that cannot be easily converted into text, while keeping similar ratios of the CRC levels (C_r). We also create a set similar to B, say B_r, that retains only resumes in C_r. C_r and B_r are used for our first task (§4.1) and second task (§4.2), respectively.

3.2 Preprocessing

The resumes collected by the HR come with several formats (e.g, DOC, DOCX, PDF, RTF). All resumes are first converted into the unstructured text format, TXT, using publicly available tools. They are then processed by our custom regular expressions designed to segment different sections in the resumes. As a results, every resume is segmented into the six sections, *Profile*, *Education*, *Work Experience*, *Activities*, *Skills*, and *Others*. Table 3 shows the ratio of resumes in each level including those sections.

	CRC1	CRC2	CRC3	CRC4	Total
WoE	98.0	98.3	97.2	97.4	98.0
EDU	96.0	95.6	96.3	96.6	96.0
PRO	94.4	94.3	94.1	94.0	94.3
ACT	40.4	43.4	47.4	40.2	42.5
SKI	37.7	36.4	33.6	41.5	36.9
OTH	32.2	32.8	30.8	37.2	32.5

Table 3: The existence ratio of each section in the CRC levels. WoE: Work Experience, EDU: Education, PRO: Profile, ACT: Activities, SKI: Skills, OTH: Others.

Most resumes consistently include the *Work Experience*, *Education*, and *Profile* sections, whereas the others are often missing. To ensure the matching quality of our regular expressions, 200 resumes are randomly checked, where 97% of them are found to have the sections segmented correctly. Finally, all resumes comprising segmented sections are saved in the JSON format for machine readability.

3.3 Annotation

2 experts with experience in recruiting applicants for CRC positions of all levels design the annotation guidelines in 5 rounds by labeling each resume with either one of the four CRC levels, CRC1−4, or *Not Qualified* (NQ), indicating that the applicant is not qualified for any CRC level. Thus, a total of 5 labels are used for this annotation. For each round, 50 randomly selected resumes from C_r in Table 2, by keeping similar ratios of the CRC levels as C_r, are labeled by those two experts with improvement to subsequent guidelines based on their agreement.

Another batch of 50 resumes are then selected for the next round and annotated based on the revised guidelines. For batches 2-5, a third person (non-expert) is added and instructed to follow the guidelines developed from prior rounds; thus, annotation is completed by three people for these rounds. Table 4 shows the Fleiss Kappa scores to estimate the inter-annotator agreement (ITA) for each round with respect to the five competence levels.

	R1	R2	R3	R4	R5
NQ	13.2	53.8	38.5	52.0	66.8
CRC1	1.3	25.0	-7.3	57.3	65.3
CRC2	9.3	39.7	41.2	5.4	33.7
CRC3	29.1	63.5	66.7	69.8	69.6
CRC4	63.4	47.9	100.0	N/A	-0.7
Overall	16.1	45.3	40.7	55.5	**60.8**

Table 4: Fleiss Kappa scores measured for ITA during the five rounds of guideline development (R1−5). No annotation of CRC4 is found in the batch used for R4. The negative kappa scores are achieved for (CRC1, R3) and (CRC4, R5) that have too few samples (≤ 2).

For R1 with no guidelines designed, poor ITA is observed with the kappa score of 16.1%. The ITA gradually improves with more rounds, and reaches the kappa score of 60.8% among 3 annotators, indicating the high quality annotation in our dataset. The followings give brief summary of the guideline revisions after each round:

Round 1 (1) Clarify qualified and not-qualified applicants, (2) Define transferable skills (e.g, general research experience vs. experiences in healthcare), (3) Define clinical settings, clinical experience, and clinical research experience (4) Set requirements by levels of academic preparation.

Round 2 (1) Revise the length of clinical experience based on levels of academic preparation and whether the degree is in a scientific/health related field or non-scientific/non-health related field, (2) Refine CRC2−4 degree requirements, years of clinical research, and clinical experience requirements, (3) Require clinical research certification for CRC4.

Round 3 (1) Update glossary examples of clinical settings, research experience, and clinical experiences with job titles, (2) Revise years of experience in clinical roles and research experience. (3) Add categorization of foreign trained doctors and bench/laboratory research personnel.

Round 4 (1) Remove clinical experience requirements from CRC2−4 and require a minimum of 1-year clinical research for those with a scientific vs. non-scientific degree, (2) Revisit laboratory scientist requirements, (3) Remove academic experience as a research assistant unless it involved over 1000 hours. Rationale: participation by semester is typically data entry or participation in a component of the research but not full engagement in a project.

Round 5 Increase the number of years required for a bench/laboratory researcher.

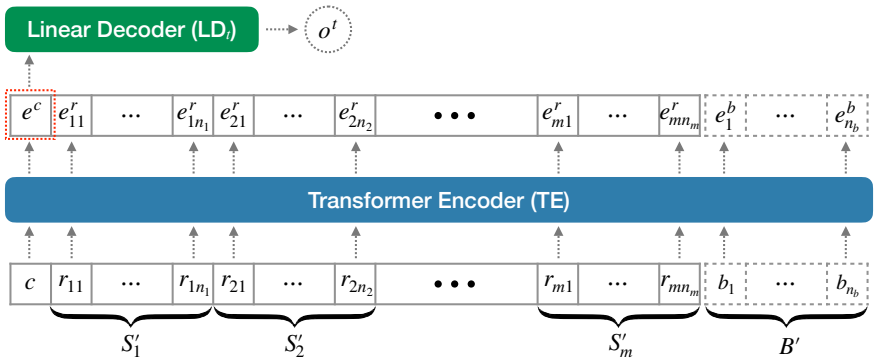

Figure 1: The whole context model using section trimming, used as baseline for T1 (§4.1.1) and T2 (§4.2.1).

During these five rounds, 250 resumes are triple annotated and adjudicated. Given the established annotation guidelines,[1] additional 3,175 resumes are single annotated and sample-checked. Thus, a total of 3,425 resumes are annotated for this study.

4 Approach

This section introduces transformer-based neural approaches to address the following two tasks:

- T1 Given a resume, decide which level of CRC positions that the corresponding applicant is suitable for (Section 4.1).
- T2 Given a resume and a CRC job description, decide whether or not the applicant is suitable for that particular job (Section 4.2).

T1 is a multiclass classification task where the labels are the five CRC levels including NQ (Table 4). This task is useful for applicants who may not have clear ideas about what levels they are eligible for, and recruiters who want to match the applicants to the best suitable jobs available to them.

T2 is a binary classification task such that even with the same resume, the label can be either positive (accept) or negative (reject), depending on the job description. This task is useful for applicants who have good ideas about what CRC levels they fit into but want to determine which particular jobs they should apply to, as well as recruiters who need to quickly screen the applicants for interviews.

4.1 Competence-Level Classification

For the competence-level classification task (T1), a baseline model that treats the whole resume as one document (§4.1.1) is compared to context-aware models using section pruning (§4.1.2), chunk segmenting (§4.1.3), and section encoding (§4.1.4).

[1]The annotation guidelines are available at our project page.

4.1.1 Whole-Context: Section Trimming

Figure 1 shows an overview of the whole context model. Let $R = \{S_1, \ldots, S_m\}$ be a resume while $S_i = \{r_{i1}, \ldots, r_{i\ell_i}\}$ is the i'th section in R where r_{ij} is the j'th token in S_i. Let N be the maximum number of input tokens that a transformer encoder can accept. Then, n_i, the max-number of tokens in S_i allowed to be input, is measured as follows:

$$T = \textstyle\sum_{\forall j} |S_j|$$
$$n_i = \min(N, T) \cdot \frac{|S_i|}{T}$$

Let $S'_i = \{r_{i1}, \ldots, r_{in_i}\}$ be the trimmed section of S_i by discarding all tokens $r_{ij} \in S_i$ ($n_i < j \leq \ell_i$). All trimmed sections are appended in order with the special token c, representing the entire resume, which creates the input list $I = \{c\} \oplus S'_1 \oplus \cdots \oplus S'_m$. I is fed into the transformer encoder (TE) that generates the list of embeddings $\{e^c\} \oplus E'_1 \oplus \ldots \oplus E'_m$, where $E'_i = \{e^r_{i1}, \ldots, e^r_{in_i}\}$ is the embedding list of S'_i, and e^c is the embeddings of c. Finally, e^c is fed into the linear decoder (LD$_t$) that generates the output vector $o^t \in \mathbb{R}^d$ to classify R into one of the competence levels (in our case, $d = 5$).

4.1.2 Context-Aware: Section Pruning

Section trimming in Section 4.1.1 allows the whole-context model to take part of every section as input. However, it is still limited because not all features necessary for the classification are guaranteed to be in the trimmed range. Moreover, this model makes no distinction between contents from different sections once $S'_{1..m}$ are concatenated. This section proposes a context-aware model to overcome those two issues by pruning tokens more intelligently and encoding each section separately so that the model learns weights for individual sections to make more informed predictions. Figure 2 shows an overview of the context-aware model using section pruning.

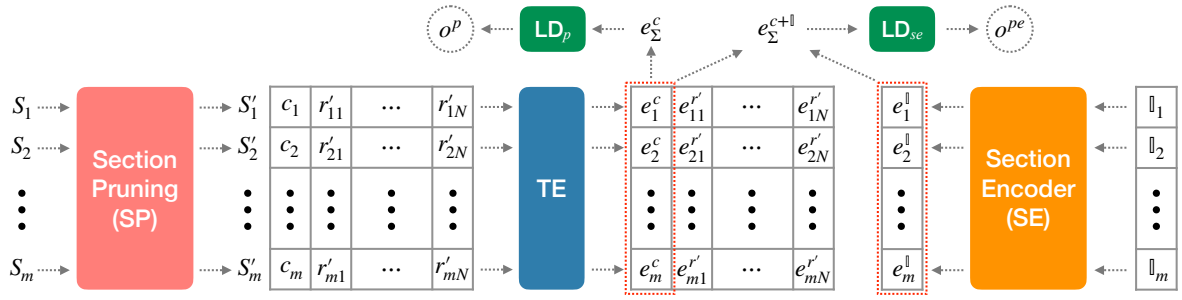

Figure 2: The context-aware model using section pruning (§4.1.2) and section encoding (§4.1.4).

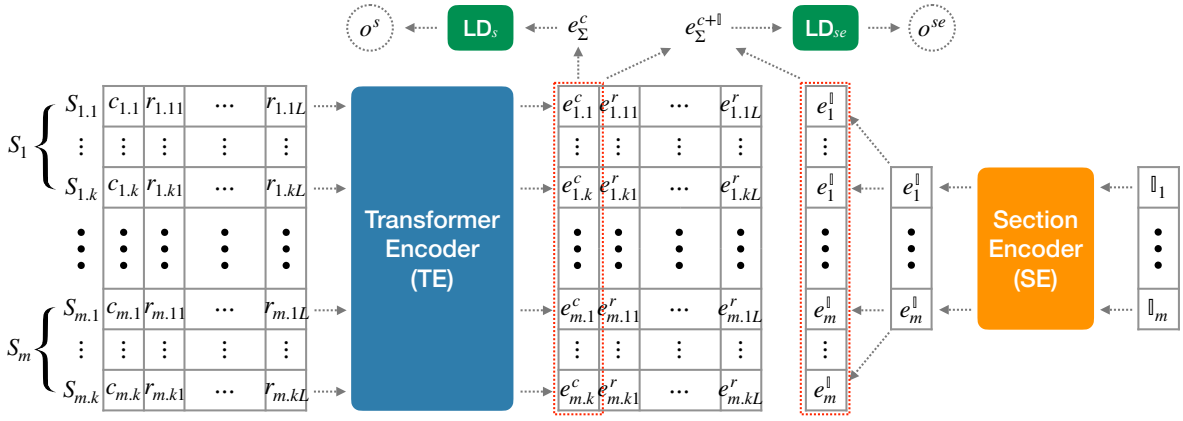

Figure 3: The context-aware model using chunk segmenting (§4.1.3) and section encoding (§4.1.4).

Given the maximum number of tokens, N, that the transformer encoder (TE) allows, any section $S_i \in R$ that contains more than N-number of tokens is pruned by applying the following procedure:

1. If $|S_i| > N$, remove all stop words in S_i.
2. If still $|S_i| > N$, remove all words whose document frequencies are among the top 5%.
3. If still $|S_i| > N$, remove all words whose document frequencies are among the top 30%.

Then, the pruned section S'_i is created for every S_i, where $S'_i \subseteq S_i$ and $|S'_i| \leq N$. Each S'_i is prepended by the special token c_i representing that section and fed into the transformer encoder (TE) that generates the list $\{e^c_i, e^{r'}_{i1}, \ldots, e^{r'}_{iN}\}$, where e^c is the embedding of c, called section embedding, and the rest are the embeddings of S'_i. Let $e^c_\Sigma = \sum_{i=1}^{m} e^c_i$, that is the sum of all section embeddings representing the whole resume. Finally, e^c_Σ is fed into the linear decoder ($\texttt{LD}_p$) that generates the output vector $o^p \in \mathbb{R}^d$ to classify R into a competence level.

4.1.3 Context-Aware: Chunk Segmenting

Section pruning in §4.1.2 preserves relevant information more than section trimming in §4.1.1; however, the model still cannot see the entire resume. Thus, this section proposes another method that uniformly segments the resume into multiple chunks and encodes each chunk separately. Figure 3 shows the context-aware model using chunk segmenting. Let $S_i = \{S_{i.1}, \ldots, S_{i.k}\}$ be the i'th section in R, where $S_{i.j}$ is the j'th chunk in S_i and $k = \lceil |S_i|/L \rceil$ given the maximum length L of any chunk so that $|S_{i.j}| = L$ for $\forall j < k$ and $|S_{i.k}| \leq L$.[2] Each chunk $S_{i.j}$ is prepended by the special token $c_{i.j}$ representing that chunk and fed into TE that generates the embedding list $E_{i.j} = \{e^c_{i.j}, e^r_{i.j1}, \ldots, e^r_{i.jL}\}$. Let $e^c_\Sigma = \sum_{\forall i \forall j} e^c_{i.j}$. Finally, e^c_Σ is fed into $\texttt{LD}_s$ that generates the output vector $o^s \in \mathbb{R}^d$ to classify R.

4.1.4 Context-Aware: Section Encoding

Chunk segmenting in §4.1.3 allows the model to see the entire resume; however, it loses information about which sections the chunks belong to. This section proposes a method to distinctively encode chunks from different sections, that can be applied to both models using section pruning (§4.1.2) and chunk segmenting. Figures 2 and 3 describe how section pruning can be applied to those two models.

Let $H = \{\mathbb{I}_1, \ldots, \mathbb{I}_m\}$ be the list of section IDs where $\mathbb{I}_i$ is the ID of the i'th section. H is then fed

[2] $S_{i.j} = \{r_{i.j1}, \ldots, r_{i.jL}\}$ and $r_{i.jp}$ is the p'th token in $S_{i.j}$, that is $r_{iq} \in S_i$ where $q = L \cdot (j-1) + p$.

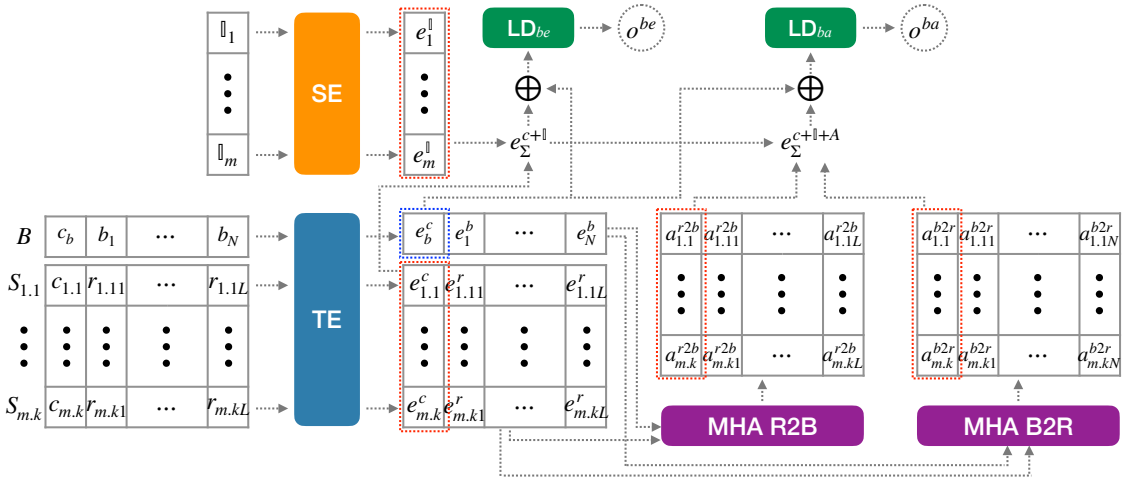

Figure 4: The context-aware models using chunk segmenting (4.1.3) + section encoding (§4.1.4) + job description embedding (§4.2.2), and multi-head attention between the resume and the job description (§4.2.3).

into the section encoder (SE), an embedding layer that learns the embedding list $E^{\mathbb{I}} = \{e^{\mathbb{I}}_1, \ldots, e^{\mathbb{I}}_m\}$ during training, where $e^{\mathbb{I}}_i$ is the embedding of $\mathbb{I}_i$. For the section pruning model in Figure 2, given $E^c = \{e^c_1, \ldots, e^c_m\}$, $F = E^c + E^{\mathbb{I}} = \{f_1, .., f_m\}$ and $e^{c+\mathbb{I}}_\Sigma = \sum_{i=1}^{m} f_i$. For the chunk segmenting model in Figure 3, given $E^c = \{e^c_{1.1}, \ldots, e^c_{m.k}\}$, $F = \{f_{1.1}, \ldots, f_{m.k}\}$ where $f_{i.j} = e^c_{i.j} + e^{\mathbb{I}}_i$. Let $e^{c+\mathbb{I}}_\Sigma = \sum_{\forall i \forall j} f_{i.j}$. Finally, $e^{c+\mathbb{I}}_\Sigma$ is fed into $\texttt{LD}_{se}$ that create $o^{pe} \in \mathbb{R}^d, o^{se} \in \mathbb{R}^d$ for the section pruning and chunk segmenting models, respectively.

4.2 Resume-to-Job_Description Matching

For the resume-to-job_description matching task (T2), the whole-context model is adapted to establish the baseline (§4.2.1), and compared to context-aware models using chunk segmenting + section encoding coupled with the job description embedding (§4.2.2), as well as multi-head attentions between the resume and the job description (§4.2.3).

4.2.1 Whole-Context: Sec./Desc. Trimming

The whole context model is similar to the one using section trimming in §4.1.1 with the additional input from the job description, illustrated as the dotted boxes in Figure 1. Let $B = \{b_1, \ldots, b_{\ell_b}\}$ be the job description where b_i is the i'th token in B. Given the max-number of tokens N that a transformer encoder can accept, the max-numbers of tokens in S_i and B, that are n_i and n_b respectively, allowed to be input are measured as followed:

$$T = \textstyle\sum_{\forall j} |S_j| + |B|$$
$$n_i = \min(N, T) \cdot {}^{|S_i|}/_{T}$$
$$n_b = \min(N, T) \cdot {}^{|B|}/_{T}$$

Let $B' = \{b_1, \ldots, b_{n_b}\}$ be the trimmed job description discarding all tokens $b_j \in B$ $(n_b < j \leq \ell_b)$. Then, the input list $I = \{c\} \oplus S'_1 \oplus \cdots \oplus S'_m \oplus B'$ is created and fed into TE that generates the embedding list $\{e^c\} \oplus E_1 \oplus \ldots \oplus E_m \oplus E_b$, where E_b is the embeddings of B'. Finally, e^c is fed into LD that generates $o^t \in \mathbb{R}^2$ to make the binary decision of whether or not R is suitable for B.

4.2.2 Context-Aware: Chunk Segmenting + Section Encoding + Desc. Embedding

The most advanced competence-level classification model using chunk segmenting (§4.1.3) and section encoding (§4.1.4) is adapted for the context-aware model with the addition of $B = \{c_b, b_1, \ldots, b_N\}$, which is fed into TE to generate the embedding list $E^b = \{e^c_b, b^b_1, \ldots, b^b_N\}$. Then, the job description embedding e^c_b is concatenated with the section encoded resume embedding $e^{c+\mathbb{I}}_\Sigma$ (§4.1.4) and fed into $\texttt{LD}_{be}$ that generates $o^{be} \in \mathbb{R}^2$.

4.2.3 Context-Aware: Multi-Head Attention

Figure 4 depicts an overview of the context-aware model using the techniques in §4.2.2 empowered by multi-head attention (Vaswani et al., 2017) between the resume R and the job description B, which allows the model to learn correlations between individual tokens in R and B, r_* and b_*, as well as the chunk and job description representations, c_*.

Let $\mathcal{E}^r \in \mathbb{R}^{\gamma \times \lambda}$ be the matrix representing R, where γ is the total number of chunks across all sections in R, $\lambda = L + 1$, and L is the max-length of any chunk. Thus, each row in $\mathcal{E}^r$ is the embedding list $E_{i.j} \in \mathbb{R}^{1 \times \lambda}$ of the corresponding chunk $S_{i,j}$. Let $\mathcal{E}^b \in \mathbb{R}^{\gamma \times \nu}$ be the matrix representing B where

$\nu = N+1$ and N is the max-length of B. Each row in $\mathcal{E}^b$ is a copy of the embedding list $E^b \in \mathbb{R}^{1\times\nu}$ in §4.2.2. Thus, every row is identical to the other rows in $\mathcal{E}^b$. These two matrices, $\mathcal{E}^r$ and $\mathcal{E}^b$, are fed into two types of multi-head attention (MHA) layers, one finding correlations from R to B (R2B) and the other from B to R (B2R), which generate two attention matrices, $\mathcal{A}^{r2b} \in \mathbb{R}^{\gamma\times\lambda}$ and $\mathcal{A}^{b2r} \in \mathbb{R}^{\gamma\times\nu}$.

The embeddings of the chunks, $\{e^c_{1.1}, \ldots, e^c_{m.k}\}$, and the section encodings, $\{e^{\mathbb{I}}_1, \ldots, e^{\mathbb{I}}_m\}$, as well as the outputs of MHA-R2B, $\{a^{r2b}_{1.1}, \ldots, a^{r2b}_{m.k}\}$, and MHA-B2R, $\{a^{b2r}_{1.1}, \ldots, a^{b2r}_{m.k}\}$, together make $F^a = \{f^a_{1.1}, \ldots, f^a_{m.k}\}$ s.t. $f_{i.j} = e^c_{i.j} + e^{\mathbb{I}}_i + a^{r2b}_{i.j} + a^{b2r}_{i.j}$. Finally, $e^{c+\mathbb{I}+A}_{\Sigma} = \sum_{\forall i \forall j} f^a_{i.j}$ is fed into $\mathtt{LD}_{ba}$ that generates $o^{ba} \in \mathbb{R}^2$ for the binary classification.

5 Experiments

5.1 Data Distributions

Table 5 shows the data split used to develop models for the competence-level classification task (T1). The annotated data in the row $\mathtt{C}_r$ of Table 2 are split into the training (TRN), development (DEV) and test (TST) sets with the ratios of 75:10:15 by keeping similar label distributions across all sets.

	TRN	DEV	TST	Total	Dist.
NQ	355	48	72	475	13.87%
CRC1	1,510	202	302	2,014	58.80%
CRC2	286	38	58	382	11.15%
CRC3	392	53	79	524	15.30%
CRC4	22	3	5	30	0.88%
Total	2,565	344	516	3,425	100.00%

Table 5: Data statistics for the competence-level classification task (T1) in Section 4.1.

70% of the data are annotated with the entry levels, CRC1 and CRC2, that is not surprising since 77.3% of the applications are submitted for those 2 levels. The ratio of CRC4 is notably lower than the application ratio submitted to that level, 6.8%, implying that applicants tend to apply to jobs for which they are not qualified. 13.9% of the applicants are NQ; thus, if our model detects even that portion robustly, it can remarkably reduce human labor.

Table 6 shows the data split used for the resume-to-job_description matching task (T2). The same ratios of 75:10:15 are applied to generate the TRN:DEV:TST sets, respectively. Note that an applicant can submit resumes to more than one CRC level. Algorithm 1 is designed to avoiding any overlapping applicants across datasets while keeping the similar label distributions (Appendix A.1).

		TRN	DEV	TST	Total	Dist.
CRC1	Y	1,279	171	257	1,707	31.84%
	N	772	100	151	1,023	19.08%
CRC2	Y	183	25	38	246	4.59%
	N	1,086	148	222	1,456	27.15%
CRC3	Y	153	21	32	206	3.84%
	N	373	46	71	490	9.14%
CRC4	Y	8	0	2	10	0.19%
	N	169	22	33	224	4.18%
Total		4,023	533	806	5,362	100.00%

Table 6: Data statistics for the resume-to-job_ description matching task (T2) in Section 4.2. Y/N: applicants whose applied CRC levels match/do not match our annotated label, respectively.

Out of the 5,362 applications, 40.5% of them match our annotation of the CRC levels, indicating that less than a half of applications are suitable for the positions they apply. The number of matches drops significantly for CRC2; only 14.5% are found to be suitable according to our labels. Too few instances are found for CRC4; only 4.3% of the applicants applying for this level match our annotation.

5.2 Models

For our experiments, the BERT base model is used as the transformer encoder (Devlin et al., 2019) although our approach is not restricted to any particular type of encoder. The following models are developed for T1 (Section 4.1):

- $\mathtt{W}_r$: Whole context model + section trimming (§4.1.1)
- P: Context-aware model + section pruning (§4.1.2)
- P⊕I: P + section encoding (§4.1.4)
- C: Context-aware model + chunk segmenting (§4.1.3)
- C⊕I: S + section encoding (§4.1.4)

The followings are developed for T2 (Section 4.2):

- $\mathtt{W}_{r+b}$: Whole context + sec./job_desc. trimming (§4.2.1)
- P⊕I⊕J: P⊕I + job_desc. embedding (≈§4.2.2)
- P⊕I⊕J⊕A: P⊕I⊕J + multi-head attention (≈§4.2.3)
- P⊕I⊕J⊕A⊖E: P⊕I⊕J - E^c (§4.1.4)
- C⊕I⊕J: C⊕I + job_desc. embedding (§4.2.2)
- C⊕I⊕J⊕A: C⊕I⊕J + multi-head attention (§4.2.3)
- C⊕I⊕J⊕A⊖E: C⊕I⊕J - E^c (§4.1.4)

The P⊕I⊕J model adapts section pruning to generate $e^{c+\mathbb{I}}_{\Sigma}$ instead of chunk segmenting in §4.2.2. For the P⊕I⊕J⊕A model, the attention matrices in §4.2.3 are reconfigured as $\mathcal{A}^{r2b}, \mathcal{A}^{b2r} \in \mathbb{R}^{m\times\nu}$ (m: the number of sections in R). These models are developed to make comparisons between those two approaches for T2. Also, the $\star\ominus$E models exclude the embedding list E^c such that $f_{i.j}$ is redefined as $f_{i.j} = e^{\mathbb{I}}_i + a^{r2b}_{i.j} + a^{b2r}_{i.j}$ in §4.2.3 to estimate the pure impact of multi-head attention.

5.3 Results

Labeling accuracy is used as the evaluation metric for all our experiments. Each model is developed three times and their average score as well as the standard deviation are reported.[3] Table 7 shows the results for T1 achieved by the models in Sec. 5.2. All context-aware models without section encoding perform significantly better, 1.5% with section pruning (P) and 3.3% with chunk segmenting (C), than the baseline model (W_r). C shows a greater improvement of 1.8% than P, implying that the additional context used in C is essential for this task. Section encoding (I) helps both P and C. As the result, C⊕I shows 4.2% improvement over W_r and also gives the least variance of 0.16.

	DEV	TST	δ
W_r	69.38 (±0.14)	69.06 (±1.56)	-
P	68.99 (±0.49)	70.58 (±0.38)	1.52
P⊕I	69.19 (±0.63)	70.87 (±0.40)	1.81
C	70.36 (±0.34)	72.35 (±0.24)	3.29
C⊕I	**70.64** (±0.41)	**73.26** (±0.16)	**4.20**

Table 7: Accuracy (± standard deviation) on the development (DEV) and test (TST) sets for T1, achieved by the models in Section 5.2. δ: delta over W_r on TST.

Table 8 shows the results for T2 achieved by the models in Section 5.2. Neither the context-aware model using section pruning (P) or chunk segmenting (C) with section encoding (⊕I) performs better than the baseline model (W_{r+b}) by simply concatenating the job description embedding (⊕J). Indeed, none of the P⊕⋆ models performs better than W_{r+b}, that is surprising given the success they depict for T1 (Table 7). However, C with multi-head attention (C⊕I⊕J⊕A) show a significant improvement of 4.6% over its counterpart, that is very encouraging.

	DEV	TST	δ
W_{r+b}	76.24 (±1.08)	77.70 (±0.59)	-
P⊕I⊕J	74.73 (±0.54)	75.60 (±1.07)	-2.1
P⊕I⊕J⊕A	75.36 (±0.57)	77.25 (±0.87)	-0.5
P⊕I⊕J⊕A⊖E	76.42 (±0.22)	77.58 (±0.95)	-0.1
C⊕I⊕J	73.85 (±0.87)	74.65 (±1.87)	-3.1
C⊕I⊕J⊕A	**76.99** (±1.10)	**79.20** (±0.26)	**1.5**
C⊕I⊕J⊕A⊖E	76.20 (±0.96)	78.49 (±0.74)	0.8

Table 8: Accuracy (± standard deviation) on the development (DEV) and test (TST) sets for T2, achieved by the models in Section 5.2. δ: delta over W_r on TST.

Multi-head attention (A) gives good improvement to P as well. Interestingly, the one excluding the embedding list (⊖E) performs slightly better than the one including it (P⊕I⊕J⊕A), implying that the embeddings from the pruned sections are not as useful once the attention is in place.

[3] Appdendix A.2 provides details of our experimental settings for the replicability of this work.

5.4 Analysis

Figure 5 shows the confusion matrix for T1's best model, C⊕I. The prediction of CRC1 shows robust performance, which has the most number of training instances (Table 5), whereas the other dimensions are mostly confused around their neighbors, often hard to distinguish even for human experts.

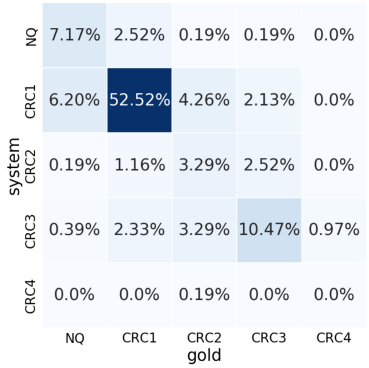

Figure 5: Confusion matrix for the best model of T1.

Figure 6 shows the confusion matrix for T2's best model, C⊕I⊕J⊕A. In general, this model shows robust performance across all dimensions.

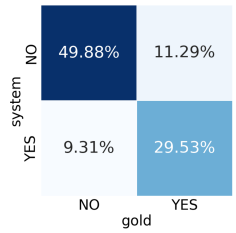

Figure 6: Confusion matrix for the best model of T2.

5.5 Error Analysis

This section provides a detailed analysis from our experts about prediction errors made by our best model in Section 5.3.

General The following observations are found as general error cases:

- Classifying foreign trained MDs and persons with PhDs with no clinical research experience to overrate them. (1) It picks up research project done in training as significant research. (2) It is unable to identify clinical research experience.

- Classifying laboratory personnel entering CRC area.
- Counting research experience: identifying dates of experience. (1) It needs to accumulate experience (e.g., CRC1: 6 months; CRC2: 2-3 years). (2) It needs implications for creating a structured entry form versus resume (3) Academic research experiences that are less than 1000 hours not counted; a semester experience not counted. (4) It needs to count paid research experience.
- Not picking up research related titles or terms such as (1) Research coordinator, research assistant, senior assistant; (2) IRB, informed consent, regulatory, specimen management, SOP, interviews, questionnaires; (3) Lab researcher: assays, immunohistochemistry.
- Not recognizing transferable skills such as clinical setting, clinical experience, and laboratory experience.
- Recognizing correct certifications. CRC positions require Clinical Research Certification but do not require CITI or CPR Certificates.
- Not distinguishing levels of preparation and associated clinical experience or research experience. Distinguishing scientific vs. nonscientific degrees for CRC1 and CRC2 is particularly important.

The following error cases are found between the adjacent pairs of CRC positions:

NQ vs. CRC1 It needs to distinguish transferable skills, clinical setting, clinical experiences.

CRC1 vs. CRC2 It needs to count for (1) Levels of education; (2) Scientific vs. non-scientific degree; (3) Clinical experience that is a must for CRC2 at lower educational levels;

CRC2 vs. CRC3 It needs to count for (1) Length of clinical research experience; (2) Foreign trained MD; (3) Laboratory personnel length of time

CRC3 vs. CRC4 (1) Foreign MD are often classified too high. (2) CRC4 needs Certification in Clinical Research

6 Conclusion

This paper proposes two novel tasks, competence-level classification (T1) and resume-description matching (T2), and provides a high-quality dataset as well as robust models using several transformer-based approaches. The accuracies achieved by our best models, 73.3 for T1 and 79.2 for T2, show a good promise for these models to be deployed in real HR systems. To the best of our knowledge, this is the first time that those two tasks are thoroughly studies, especially with the latest transformer architectures. We will continuously explore to improve these models by integrating expert's knowledge.

References

Emil St. Chifu, Viorica R. Chifu, Iulia Popa, and Ioan Salomie. 2017. A System for Detecting Professional Skills from Resumes Written in Natural Language. In *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing*, ICCP'17, pages 189–196.

Yu Deng, Hang Lei, Xiaoyu Li, and Yiou Lin. 2018. An Improved Deep Neural Network Model for Job Matching. In *Proceedings of the International Conference on Artificial Intelligence and Big Data*, ICAIBD'18.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Melanie A. Myers. 2019. Healthcare Data Scientist Qualifications, Skills, and Job Focus: A Content Analysis of Job Postings. *Journal of the American Medical Informatics Association*, 26(5):383–391.

S. Nasser, C. Sreejith, and M. Irshad. 2018. Convolutional Neural Network with Word Embedding Based Approach for Resume Classification. In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*, pages 1–6.

Luiza Sayfullina, Eric Malmi, Yiping Liao, and Alexander Jung. 2017. Domain adaptation for resume classification using convolutional neural networks. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 82–93. Springer.

Darin Stewart. 2019. Understanding Your Customers by Using Text Analytics and Natural Language Processing. *Gartner Research*, G00373854.

Rogelio Valdez-Almada, Oscar M. Rodriguez-Elias, Cesar E. Rose-Gomez, Maria D. J. Velazquez-Mendoza, and Samuel Gonzalez-Lopez. 2018. Natural Language Processing and Text Mining to Identify Knowledge Profiles for Software Engineering Positions Generating Knowledge Profiles from Resumes. In *Proceedings of the International Conference in Software Engineering Research and Innovation*, CONISOFT'18.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA. Curran Associates Inc.

A. Zaroor, M. Maree, and M. Sabha. 2017. JRC: A Job Post and Resume Classification System for Online Recruitment. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 780–787.

A Appendices

A.1 Spliting Algorithm for T2

Algorithm 1 is to split the TRN/DEV/TST sets for T2 (Table 6) without overlapping applicants across them while keeping the label distributions. The key idea is to split the data by targeted label distributions but with a smaller training set ratio than the original one. If there are overlapping applicants, then it puts all of the overlaps into the training set so that the training set ratio will be large enough to be close to the targeted training set ratio while the label distributions are still kept in a great extent.

Algorithm 1: Splitting Algorithm for T2

```
Result: The splitted dataset for T2
Initialize a random training set ratio T_i smaller than
  the targeted training and evaluation ratio T_t;
while True do
    Split the training and evaluation set by T_i based
      on the ratio R of positions applied and
      annotated matching results;
    if There are overlap resumes between training
      and evaluation set then
        Put all overlap resumes into the splitted
          training set;
        Compute the new training ratio T_n;
        if T_n is not closed to T_t then
            Adjust T_i based on the relation between
              T_n and T_t;
            Continue;
        else
            Split the evaluation set into the
              development and test set based on R;
            Return the splitted set;
        end
    else
        if T_i is not closed to T_t then
            Adjust T_i based on the relation between
              T_i and T_t;
            Continue;
        else
            Split the evaluation set into the
              development and test set based on R;
            Return the splitted set;
        end
    end
end
```

A.2 Experimental Settings

Table 9 shows the hyper-parameters used for each model (Section 5.2). For chunk segmenting in Section 4.1.3, let k_i be the number of chunks in the i'th section, then $K = \sum_{i=1}^{m} k_i$ is the total number of chunks in R. To utilize the GPU memory wisely, resumes with the same K are put to the same batch and different batches are trained with different batch sizes based on K and GPU memory to maximum the GPU usage. Different seeds are used when developing models for three times.

Model	L	GAS	BS	LR	E	T	PS
W_r/W_{r+b}	512	2	5	2e-05	20	1-3h	109M
P/P⊕I	512	2	3	2e-05	20	4-6h	109M
C/C⊕I	128	1	1,2,4	2e-05	20	4-6h	109M
P⊕I⊕⋆	512	2	3	2e-05	20	6-8h	112M
C⊕I⊕⋆	128	1	1,2,4	2e-05	20	6-8h	112M

Table 9: Hyperparameters. L: TE input length; GAS: gradient accumulation steps; BS: batch size; LR: learning rate; E: number of training epochs; T: approximate training time(h: hours); PS: approximate models training parameters size.

A.3 Analysis on Section Pruning

Section pruning is used to discard insignificant tokens in order to meet the limit of input size required by the transformer encoder (Section 4.1.2). Tables 10 and 11 show the section lengths before and after section pruning, respectively. These tables show that section pruning can noticeably reduce the maximum and average lengths of the sections.

Section	Average (±stdev)	Max	Ratio
Profile	100.65 (±215.75)	2139	94.93%
Skills	60.70 (±102.61)	1157	98.95%
Work Experience	314.61 (±316.61)	3605	80.26%
Education	174.30 (±289.37)	3662	89.50%
Other	77.41 (±145.40)	2184	98.34%
Activities	168.09 (±289.40)	3967	91.13%

Table 10: Section lengths before section pruning (Section 4.1.2). Average/Max: the average and max lengths of input sections. Ratio: the ratios of input sections that are under the max-input length restricted by the transformer encoder.

Section	Average(±stdev)	Max	Ratio
Profile	77.95(±127.70)	1514	99.60%
Skills	55.59 (±70.36)	546	99.93%
Work Experience	232.63 (±168.84)	2099	98.98%
Education	129.91 (±165.06)	1755	98.81%
Other	72.19 (±108.80)	1468	99.38%
Activities	125.71 (±57.74)	1514	99.13%

Table 11: Section lengths after section pruning (Section 4.1.2). Average/Max: the average and max lengths of input sections. Ratio: the ratios of input sections that are under the max-input length restricted by the transformer encoder.