Template-based Recruitment Email Generation for Job Recommendation

Qiuchi Li and **Christina Lioma**
Department of Computer Science
University of Copenhagen
{qiuchi.li, c.lioma}@di.ku.dk

Abstract

Text generation has long been a popular research topic in NLP. However, the task of generating recruitment emails from recruiters to candidates in the job recommendation scenario has received little attention by the research community. This work aims at defining the topic of automatic email generation for job recommendation, identifying the challenges, and providing a baseline template-based solution for Danish jobs. Evaluation by human experts shows that our method is effective. We wrap up by discussing the future research directions for better solving this task.

1 Introduction and Prior Work

Recruitment email generation is a crucial step in the overall job recruitment process. When recruiters find suitable candidates for a job, they need to write emails to contact the candidates, explaining why they are fit for the job and inviting them to apply. The job market produces a huge number of job postings on a daily basis, which results in a significant amount of human labor required to write recruitment emails. The recruiters are in urgent need of an automatic approach to compose these emails, to reduce their efforts and increase productivity.

The core challenges facing the task are two-fold: **persuasiveness** and **personalization**. First, the emails should contain sufficient reasons to convince the candidate that their qualifications match the requirements of the position, and illustrate that the position meets the candidate's expectations. Second, personalized emails should let the job seekers feel that the recruiters pay special attention to them and in turn motivate them to apply. In an interview of recruiters of varying recruitment experience, recruiters believe that proper personalization on the recruitment emails could boost candidate positive response rate, and admit that the current emails, largely composed based on fixed templates, are not personalized and persuasive enough (Bogers and Kaya, 2021).

Recruitment email generation can be seen as a task-oriented text generation problem. One needs to extract information from the input job description and candidate profile, and generate the recruitment email accordingly. A large body of research on emails focuses on the analysis side, typically automatic detection of phishing emails by extracting features (Basnet et al., 2008; Verma and El Aassal, 2017; Yu et al., 2009). In the realm of email synthesis, email subject line generation (Xue et al., 2020; Zhang and Tetreault, 2019) has been studied. For generation of the main email body, attempts have been made on fake email generation for malicious purposes (Das and Verma, 2019; Baki et al., 2017) based on a two-step pipeline (Chen and Rudnicky, 2014a,b) for email synthesis. As far as we know, generating recruitment emails in job recruitment is an unexplored task in literature, despite its practical significance and challenging nature.

We therefore formally formulate the task of recruitment email generation and provide a baseline template-based system for it by extending Chen and Rudnicky (2014a)'s approach. In particular, we randomly generate an email from a library of different pre-defined components, and fill the motivational sentence with certain slots by combining matched skills and occupations extracted from the job and candidate in question. We conduct a user study to evaluate the quality of the generated text and examine if it can save recruiters' time in writing emails. The results show that recruiters under test are overall satisfied with the generated emails, and the time spent on writing emails for each candidate is significantly reduced.

The rest of the paper is organized as follows. We define the problem formally in Section 2. We run a pilot study to examine the simple end-to-end neural generation algorithm in Section 3 and elicit the need for a fine-grained synthesizing approach. Our template-based approach is described in Section 4, and the user evaluation is reported and discussed in Section 5. We conclude and point out future directions in Section 6.

2 Task Definition and Notations

Recruitment email generation calls for generating an email based on an input the job and a candidate . It requires us to extract the matched information from different job and candidate fields, and convert it to natural language expressions in the email. Typically, a job posting contains its title, company name and a textual description. A candidate profile has a headline, a list of keywords, a list of preferred job titles, the candidate's work experience and educational experience, and the candidate's resume text.

The data we work with comes from Jobindex[1], one of Scandinavia's largest job portals and recruitment agencies. In the Jobindex system, the abovementioned fields are the main source of information for jobs and candidates. The majority of the jobs and candidate profiles are in Danish. Please refer to Fig. 1 for an example of a pair of job and candidate, and the real recruitment email written by a human recruiter. The personal information has been removed from the example and all texts are translated to English.

3 Pilot Study: end-to-end neural generation

The most intuitive way to handle the recruitment email generation task is to generate the whole email in a fully end-to-end fashion by concatenating the job and company text. We conduct a pilot study to examine whether this setting works in practice.

Model. We build a Transformer encoder-decoder, the state-of-the-art architecture for natural language generation, to support a sequence-to-sequence generation of recruitment email. The job and candidate texts are fed to the encoder side, and the recruitment email is generated in a token-by-token auto-regressive manner. The Transformer structure enables us to load the weights of the pre-trained Danish language model, Danish-BERT [2]. Rothe et al. (2020) conducted a comprehensive comparison of different strategies to use the pre-trained BERT weights for sequence-to-sequence generation, and found that the *bert2bert* setting can yield robust performance across different text generation tasks. Therefore, we follow this setting and initialize the weights for both encoder and decoder with the Danish-BERT checkpoint. Please refer to Rothe et al. (2020) for further details. We claim that this model may not be the state-of-the-art for Danish text generation, since Danish GPT is publicly available [3]. However, it is a strong-performing model due to its Transformer architecture, and its generated text is representative of the state-of-the-art neural generation models.

Data Preprocessing. We focus on generating Danish emails from Danish jobs and candidate profiles. We mine (job, candidate, email) triplets from the Jobindex database. We filter out samples with non-Danish text, too short job descriptions and empty job titles. A total number of 317566 samples are obtained. Due to the limitation on input length by the model, we concatenate the summary of job description and the headline, job titles, education experience and work experience of candidate. Different fields are split by the periods. For recruitment emails, we only take their main bodies. We replace specific job title and company name with special tokens "[job]" and "[cpy]" respectively.

Training and Evaluation. Our Transformer encoder-decoder has 12 layers and 12 attention heads and a maximum length of 512 tokens on both encoder and decoder sides. The embedding dimension and hidden dimension are set to 768 and 3072, respectively. We split all samples into training, validation and test set at a 7:2:1 ratio. We perform mini-batch learning to train the model with a batch size of 32. The average cross-entropy loss over all tokens at the decoder side is used as the loss function. We train the model on the training set for a maximum number of 5 epochs with Adam optimizer, and stop training when the validation loss stops dropping.

We use bilingual evaluation understudy (BLEU) as the quantitative metric for the generated message. The BLEU scores are calculated for individual translated segments, by comparing them with a set of good quality reference translations. The average scores over the whole test corpus are computed as an estimation of the overall quality.

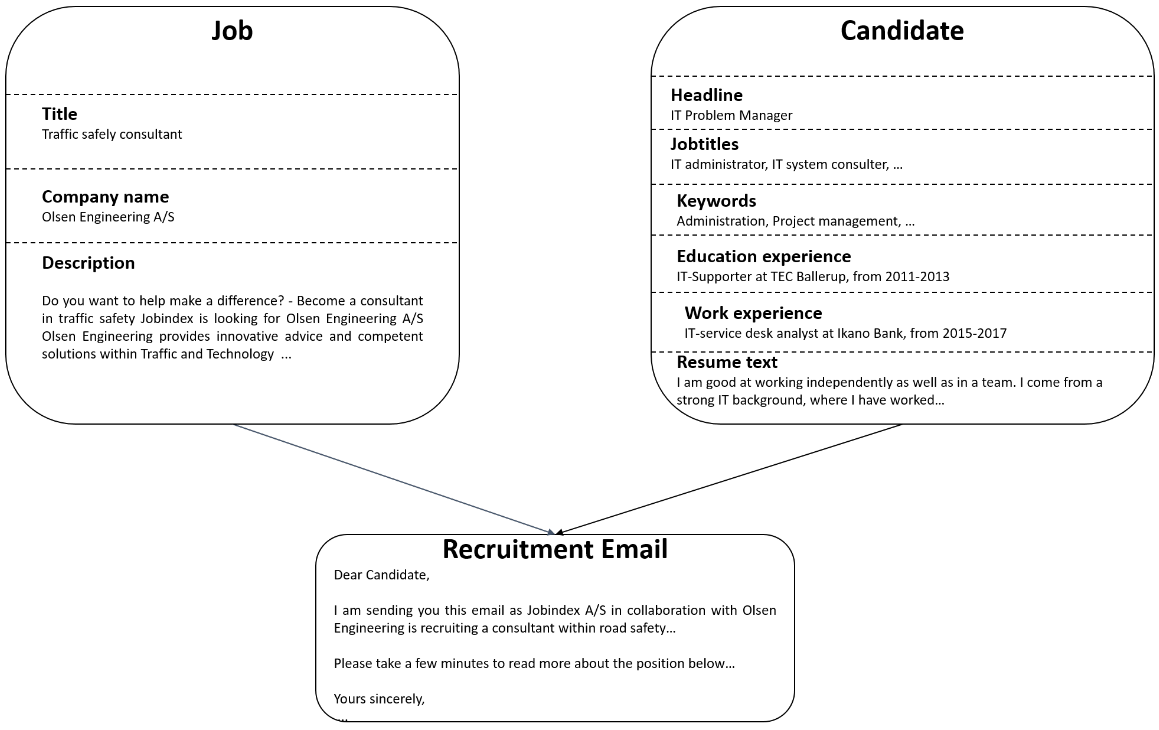
Result. We obtained a BLEU score of 23.2 on test samples for the generated texts, which is far from the SOTA neural text generation system for Danish (32.5 - 33.8) (Fan and Gardent, 2020). This

[1]https://www.jobindex.dk/

[2]https://huggingface.co/Maltehb/danish-bert-botxo

[3]https://huggingface.co/flax-community/dansk-gpt-wiki

Figure 1: Example of a real recruitment email for the given job and candidate pair. The texts are translated from Danish to English.



Title
Traffic safely consultant

Company name
Olsen Engineering A/S

Description
Do you want to help make a difference? - Become a consultant in traffic safety Jobindex is looking for Olsen Engineering A/S Olsen Engineering provides innovative advice and competent solutions within Traffic and Technology ...

Headline
IT Problem Manager

Jobtitles
IT administrator, IT system consulter, ...

Keywords
Administration, Project management, ...

Education experience
IT-Supporter at TEC Ballerup, from 2011-2013

Work experience
IT-service desk analyst at Ikano Bank, from 2015-2017

Resume text
I am good at working independently as well as in a team. I come from a strong IT background, where I have worked...

Dear Candidate,

I am sending you this email as Jobindex A/S in collaboration with Olsen Engineering is recruiting a consultant within road safety...

Please take a few minutes to read more about the position below...

Yours sincerely,
...

reveals that the generated emails have a limited overall quality.

Additionally, manual inspection shows that the auto-generated emails have low extent of personalization and persuasiveness. The generated motivational sentences usually do not contain specific reasons of match and are prone to grammatical errors. See Fig. 2 for concrete examples. We posit that it is because the motivational sentences are usually located in the middle of email body, embedded in texts primarily generated from templates. Also, over 80% of the motivational sentences are devoid of case-specific information such as the matched skills or occupations. Therefore, an end-to-end generation system is capable of learning these recurring template-based texts quite well, but poor at learning case-specific motivational sentences.

4 Proposed Methodology

The evaluation result of end-to-end neural generation indicates the need for a finer-grained generation system. Instead of generating the email as a whole, a special module should be developed to generate case-specific information, such as the motivational sentence, from the input job and candidate texts. We are inspired by the two-step email synthesis (Chen and Rudnicky, 2014a) to generate recruitment emails: email structure generation and slot filling. Specifically, we randomly generate an email template from a pre-constructed library of email components. The slots of the template indicate case-specific fields, and are then filled by extracting information from the matched job and candidate. The overall process of the system is shown in Fig. 3.

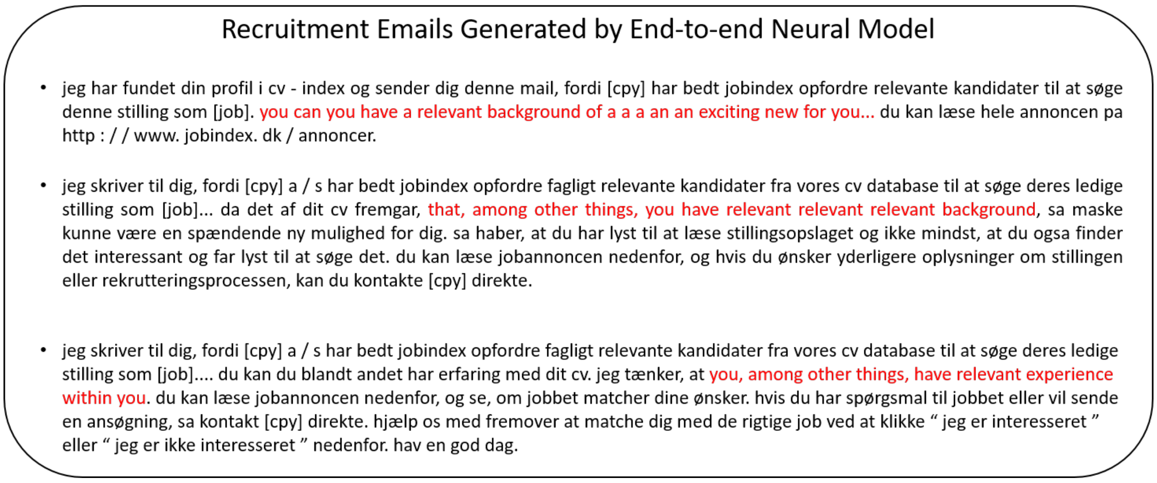
Potentially, our approach is superior to end-to-end generation in three aspects: 1) recruiters follow a similar process to write emails in a real-case scenario, so the generated emails are more likely to be accepted by and benefit recruiters; 2) the algorithm has a better control of the generated content, and ensures a certain extent of grammatical correctness and readability; 3) by explicitly composing case-specific information, the algorithm enhances personalization and persuasiveness of the generated recruitment emails.

In the following text, we introduce the implementation details of our template-based generation system.

4.1 Template Parsing

Template data cleaning. We get a total number of 553 raw templates from Jobindex system. We remove the non-Danish templates and templates for

Figure 2: Examples of generated emails by the end-to-end neural model. The generated motivational sentences are translated in English and colored in red.



Recruitment Emails Generated by End-to-end Neural Model

- jeg har fundet din profil i cv - index og sender dig denne mail, fordi [cpy] har bedt jobindex opfordre relevante kandidater til at søge denne stilling som [job]. you can you have a relevant background of a a a an an exciting new for you... du kan læse hele annoncen pa http : / / www. jobindex. dk / annoncer.
- jeg skriver til dig, fordi [cpy] a / s har bedt jobindex opfordre fagligt relevante kandidater fra vores cv database til at søge deres ledige stilling som [job]... da det af dit cv fremgar, that, among other things, you have relevant relevant relevant background, sa maske kunne være en spændende ny mulighed for dig. sa haber, at du har lyst til at læse stillingsopslaget og ikke mindst, at du ogsa finder det interessant og far lyst til at søge det. du kan læse jobannoncen nedenfor, og hvis du ønsker yderligere oplysninger om stillingen eller rekrutteringsprocessen, kan du kontakte [cpy] direkte.
- jeg skriver til dig, fordi [cpy] a / s har bedt jobindex opfordre fagligt relevante kandidater fra vores cv database til at søge deres ledige stilling som [job].... du kan du blandt andet har erfaring med dit cv. jeg tænker, at you, among other things, have relevant experience within you. du kan læse jobannoncen nedenfor, og se, om jobbet matcher dine ønsker. hvis du har spørgsmal til jobbet eller vil sende en ansøgning, sa kontakt [cpy] direkte. hjælp os med fremover at matche dig med de rigtige job ved at klikke " jeg er interesseret " eller " jeg er ikke interesseret " nedenfor. hav en god dag.

other purposes from the collection, and the remaining ones are classified into company-specific and general templates. Company-specific templates are provided by some companies to the recruiters with the request that these templates are strictly used when generating recruitment emails for these specific companies. General templates are used for all other cases. Eventually, 270 general templates and 81 specific templates remain in the database.

Manual annotation. We define a list of email components (or fields) according to their functions based on manual inspection. They are mainly categorized into **functional**, **case-specific** and **auto-fill** fields. Functional fields are indicative of email structure, and do not entail case-specific information. Case-specific fields contain information specific to the matched pair of job and candidate, such as the job title, company name, motivational sentences expressing why the candidate is fit for the job, and so on. Some case-specific fields are automatically filled by the Jobindex system, and they are referred to as auto-fill fields instead. A total number of 54 email components are defined.

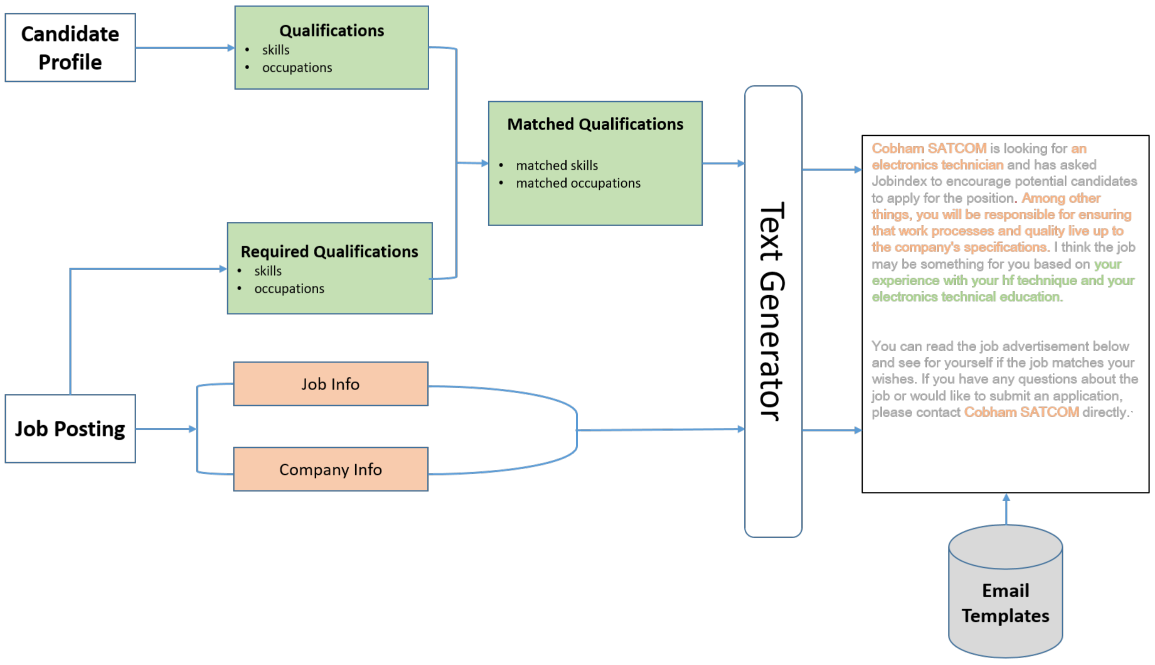
For each template in the database, we manually annotate the text by inserting HTML tags before and after a certain text segment to indicate its function in the email.

Parsing. We parse the annotated templates to construct the email component library, essentially a dictionary of (component_name, content list) pairs. The annotated templates are each parsed in a recursive fashion by an HTML parser. Starting from the root element of the whole template, the parser performs the same operations for each element: identify its child elements, process each the child element, append the processed child element content to the dictionary, and replace the child element with the marker "**[% f %]**". As such, the parser navigates all HTML tags in the template and adds their contents to the corresponding component content list. In the text content of each component, its child components are all masked with "**[% f %]**" tags as desired. The output of the algorithm is the top-level structure of the email templates, which are stored in the dictionary as values of "**skeleton_follower**" and "**skeleton_non_follower**" for followers and for non-followers, respectively. Followers are candidates who follow a company in their profile, and non-followers are candidates who do not follow a company. After parsing all templates, we obtain a list of contents for each pre-defined email component. Examples of extracted email components are shown in Fig. 4. It is worth noting that nested structures universally appear among functional components, i.e. the textual content of one component may contain other components.

4.2 Baseline Template-based Email Generation Algorithm

We build an email generation system based on the constructed email component library. The algorithm randomly generates a template from the constructed library, based on whether the database contains the company name and whether the candidate is a follower of the company. Then,

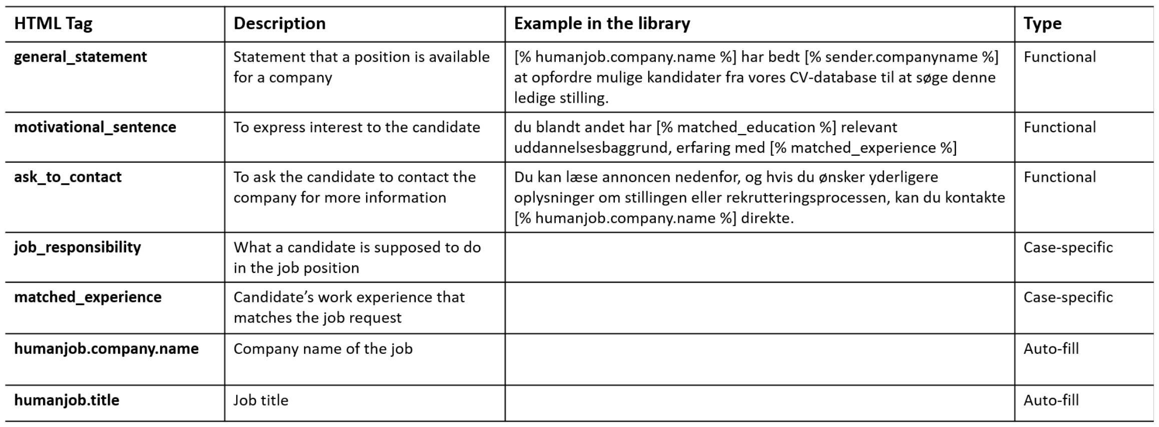
Figure 3: Diagram of the email generation system. The final generated messages are the main body of a real recruitment email, translated to English for this example.



Cobham SATCOM is looking for an electronics technician and has asked Jobindex to encourage potential candidates to apply for the position. Among other things, you will be responsible for ensuring that work processes and quality live up to the company's specifications. I think the job may be something for you based on your experience with your hf technique and your electronics technical education.

You can read the job advertisement below and see for yourself if the job matches your wishes. If you have any questions about the job or would like to submit an application, please contact Cobham SATCOM directly.

Figure 4: The HTML tag, description and example of email components, extracted from the template database.



HTML Tag	Description	Example in the library	Type
general_statement	Statement that a position is available for a company	[% humanjob.company.name %] har bedt [% sender.companyname %] at opfordre mulige kandidater fra vores CV-database til at søge denne ledige stilling.	Functional
motivational_sentence	To express interest to the candidate	du blandt andet har [% matched_education %] relevant uddannelsesbaggrund, erfaring med [% matched_experience %]	Functional
ask_to_contact	To ask the candidate to contact the company for more information	Du kan læse annoncen nedenfor, og hvis du ønsker yderligere oplysninger om stillingen eller rekrutteringsprocessen, kan du kontakte [% humanjob.company.name %] direkte.	Functional
job_responsibility	What a candidate is supposed to do in the job position		Case-specific
matched_experience	Candidate's work experience that matches the job request		Case-specific
humanjob.company.name	Company name of the job		Auto-fill
humanjob.title	Job title		Auto-fill

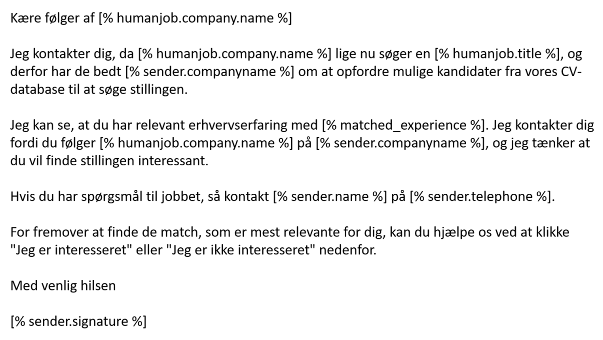
we extract the matched skills and occupations from the input job and candidate, and convert them to a coherent expression for the motivational sentence explaining why the candidate fits the job.

Random template generation. Two issues should be accounted for when generating a proper template for a matched (job, candidate) pair. First, some companies prefer writing recruitment emails based on company-specific templates, so the system should always use a company template where possible. Second, followers should be contacted with slightly different emails to stress that the candidate follows the company.

The algorithm works as follows. First, we search for templates with the company name of the job posting. If company-specific templates can be found, then we randomly choose a template from the matched company templates. If it returns an empty list, then we randomly generate a template from the email component library. We start with a randomly selected follower skeleton or a non-follower skeleton according to whether the company is in the candidate's following list. Then, we navigate through all unfilled slots in an iterative fashion. For each slot, we randomly pick a text content from all its respective candidates in the library. Since the text expression may contain other unfilled slots, this process goes in an iterative fashion until no functional component tags appear in the templates.

To this end, we have obtained a template for the recruitment email. An example can be seen in Fig. 5.

Figure 5: A randomly generated template for non-followers.


Kære følger af [% humanjob.company.name %]

Jeg kontakter dig, da [% humanjob.company.name %] lige nu søger en [% humanjob.title %], og derfor har de bedt [% sender.companyname %] om at opfordre mulige kandidater fra vores CV-database til at søge stillingen.

Jeg kan se, at du har relevant erhvervserfaring med [% matched_experience %]. Jeg kontakter dig fordi du følger [% humanjob.company.name %] på [% sender.companyname %], og jeg tænker at du vil finde stillingen interessant.

Hvis du har spørgsmål til jobbet, så kontakt [% sender.name %] på [% sender.telephone %].

For fremover at finde de match, som er mest relevante for dig, kan du hjælpe os ved at klikke "Jeg er interesseret" eller "Jeg er ikke interesseret" nedenfor.

Med venlig hilsen

[% sender.signature %]

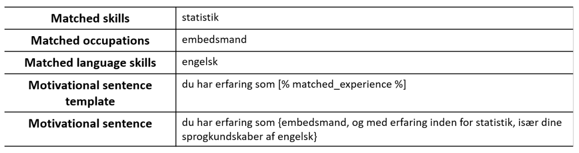
Template filling. We compose the motivational sentence based on the matched qualifications of the candidate, including skills and occupations. For this purpose, we construct a dictionary of skills and occupations from the ESCO (European Skills, Competences, Qualifications and Occupations) database[4]. ESCO has a 4-level category annotation for skills and occupations in different languages. We include both skills and occupations as dictionary keys in English and Danish. We also add the list of IT-related skills from Microsoft Skills library[5] into the dictionary. A total number of 129474 skills and 46060 occupations are collected.

Based on the dictionary, we extract skills and occupations from the matched pair of job and candidate texts and obtain their intersections. We first use a built-in name entity recognition (NER) algorithm in the spaCy[6] package to identify text spans that may contain skills and occupations, and check whether each is a skill or an occupation. The matched skills and occupations are obtained by taking the set intersections. We further split the language skills from the matched skills set.

Simple rules are then applied to convert the matched skills, occupations and language skills to a coherent expression as explanations of why the candidate fits the job. The skills and occupations are respectively concatenated with the "{}, {} og {}" pattern, and conjunctive expressions are added to combine the expressions for skill and occupation match. It is further appended by the language skill match expression, if non-Danish skills are detected in both the job and candidate. The whole expression is inserted to the motivation sentence template with modifications of the pre-slot expressions to ensure coherence. When multiple slots appear in a motivational sentence template, they are inserted with matched skills and matched occupations, respectively. As a back-up, a general text expression will be randomly chosen from a list of contents in case of no matched qualifications.

Please see Fig. 6 for a visual illustration of how the matched qualifications are converted to a motivational expression.

Figure 6: An example of composed motivational sentence from matched skills, occupations and language skills.


Matched skills	statistik
Matched occupations	embedsmand
Matched language skills	engelsk
Motivational sentence template	du har erfaring som [% matched_experience %]
Motivational sentence	du har erfaring som {embedsmand, og med erfaring inden for statistik, især dine sprogkundskaber af engelsk}

Text post-processing. We automatically post-process the generated email in the final step. Specifically, if multiple punctuations appear in a row, we keep only the last punctuation. We also make sure the first letter of a sentence is upper-cased. Finally, we correct the spacing errors between words in a sentence, between sentences, and between paragraphs. This gives the final output message produced by the template-based recruitment email generation algorithm.

5 Evaluation

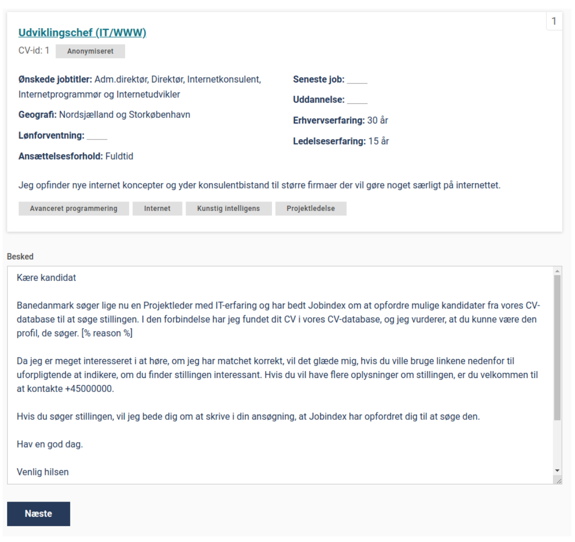
We conduct an offline evaluation with real expert recruiters to evaluate the performance of the template-based email generation algorithm. The quantitative measures for natural language generation are not employed, since the template-based approach may produce significantly different structures from the real recruitment emails and still be reasonably good. Indeed, it remains an open question to design quantitative measures for evaluating the generated recruitment emails, as the judgments on the persuasiveness or personalization of the email vary from person to person.

We randomly sample a collection of truly matched (job, candidate) pairs from the Jobindex database. They are randomly assigned to each of a group of 10 recruiters. The pairs are randomly split

[4]https://esco.ec.europa.eu/

[5]https://learn.microsoft.com/en-us/

[6]https://spacy.io/



1

Udviklingschef (IT/WWW)

CV-id: 1 Anonymiseret

Ønskede jobtitler: Adm.direktør, Direktør, Internetkonsulent, Internetprogrammør og Internetudvikler

Geografi: Nordsjælland og Storkøbenhavn

Lønforventning: ____

Ansættelsesforhold: Fuldtid

Seneste job: ____

Uddannelse: ____

Erhvervserfaring: 30 år

Ledelseserfaring: 15 år

Jeg opfinder nye internet koncepter og yder konsulentbistand til større firmaer der vil gøre noget særligt på internettet.

Avanceret programmering | Internet | Kunstig intelligens | Projektledelse

Besked

Kære kandidat

Banedanmark søger lige nu en Projektleder med IT-erfaring og har bedt Jobindex om at opfordre mulige kandidater fra vores CV-database til at søge stillingen. I den forbindelse har jeg fundet dit CV i vores CV-database, og jeg vurderer, at du kunne være den profil, de søger. [% reason %]

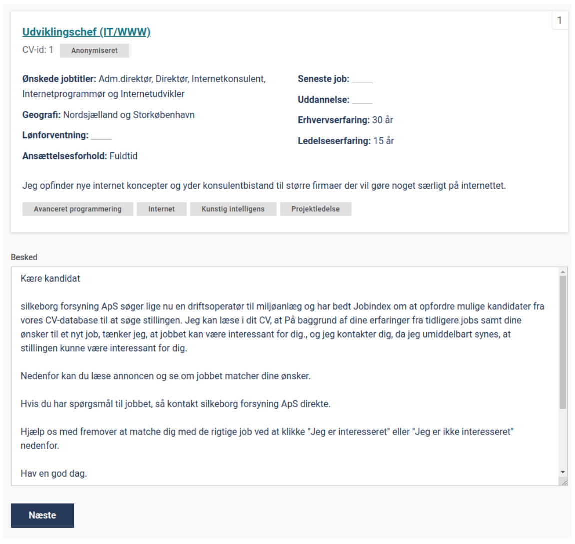
Da jeg er meget interesseret i at høre, om jeg har matchet korrekt, vil det glæde mig, hvis du ville bruge linkene nedenfor til uforpligtende at indikere, om du finder stillingen interessant. Hvis du vil have flere oplysninger om stillingen, er du velkommen til at kontakte +45000000.

Hvis du søger stillingen, vil jeg bede dig om at skrive i din ansøgning, at Jobindex har opfordret dig til at søge den.

Hav en god dag.

Venlig hilsen

(a) Unhelped case.



1

Udviklingschef (IT/WWW)

CV-id: 1 Anonymiseret

Ønskede jobtitler: Adm.direktør, Direktør, Internetkonsulent, Internetprogrammør og Internetudvikler

Geografi: Nordsjælland og Storkøbenhavn

Lønforventning: ____

Ansættelsesforhold: Fuldtid

Seneste job: ____

Uddannelse: ____

Erhvervserfaring: 30 år

Ledelseserfaring: 15 år

Jeg opfinder nye internet koncepter og yder konsulentbistand til større firmaer der vil gøre noget særligt på internettet.

Avanceret programmering | Internet | Kunstig intelligens | Projektledelse

Besked

Kære kandidat

silkeborg forsyning ApS søger lige nu en driftsoperatør til miljøanlæg og har bedt Jobindex om at opfordre mulige kandidater fra vores CV-database til at søge stillingen. Jeg kan læse i dit CV, at På baggrund af dine erfaringer fra tidligere jobs samt dine ønsker til et nyt job, tænker jeg, at jobbet kan være interessant for dig., og jeg kontakter dig, da jeg umiddelbart synes, at stillingen kunne være interessant for dig.

Nedenfor kan du læse annoncen og se om jobbet matcher dine ønsker.

Hvis du har spørgsmål til jobbet, så kontakt silkeborg forsyning ApS direkte.

Hjælp os med fremover at matche dig med de rigtige job ved at klikke "Jeg er interesseret" eller "Jeg er ikke interesseret" nedenfor.

Hav en god dag.

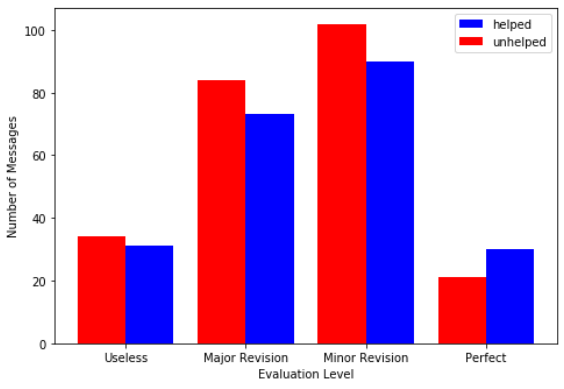
(b) Helped case.

Figure 7: Screenshots of the interface in (a) "unhelped" and (b) "helped" case.

into 5 sessions of 10 pairs each. Each recruiter is randomly assigned with "helped" tasks (with generated texts) and "unhelped" tasks (with generated templates). Each task is "helped" for half of the recruiters and "unhelped" for the other half. The interface contained pairs of matched job postings and candidate profiles, and an interface is provided to the recruiter with either pre-generated template ("unhelped") or the email ("helped"). For the email, the inserted case-specific information is wrapped by brackets "{ }". A checkbox is also presented to the recruiters for providing a 4-level judgment on the quality of the email or template (4 = Perfect, 3 = Minor Revision, 2 = Major Revision, 1 = Useless). The judgment is on language quality for templates and on both language quality and information correctness for emails. The recruiters are asked to click a button before and after writing the recruitment email, so that the difference between the recorded time stamps are the time spent on it. The screenshots of the interface are shown in Fig. 7.

Result on text quality. We have collected the results of 224 helped tasks (with generated emails by our algorithm) and 241 unhelped tasks (with our generated templates by our algorithm). The recruiters' annotations on the quality of the generated texts are shown in Fig. 8. In terms of average ratings, we obtained a value of 2.53 for generated emails and 2.46 for generated templates for the evaluation of the new template-based system. Over 50% of the generated texts (53.58% for emails and 51.03% for templates) are satisfactory, requiring minor or no edits. It is interesting to see that more recruiters are satisfied with the generated texts than the templates. This implies that the generated motivational sentence is exceptionally helpful for recruiters. In the same time, there is still room for the text quality to improve, and neural method for generating case-specific information is a feasible direction.

Figure 8: Bar plots of annotations for the offline-evaluation in Apr 2022. The red bars are the recruiters' judgments on the quality of generated templates, and blue bars are the recruiters on the quality of generated messages.



Result on time cost. We computed the average time a recruiter spent on writing the message in the unhelped and helped case, respectively. The time difference between the two scenarios is the reduction of recruiters' time by our template-based automatic text generation system. As a consequence, recruiters spent an average of 178.99s on each matched pair in the helped case, and an average of 235.63s on each matched pair in the unhelped case. On average, a recruiter spends 56.64s shorter on writing an email with automatically generated messages than merely on top of the provided templates. In order to remove the discrepancy across individual recruiters, we zero-averaged the time for each recruiter and recalculated the time difference. A closer but still remarkable gap between the times (51.31s) is observed. It shows that this simple template-based system can significantly save recruiters time on writing recruitment emails.

6 Conclusion

We have proposed recruitment email generation for job recruitment, a novel task in text generation. We demonstrate its challenge and significance in the pilot study and a user study of our template-based approach. The challenge mainly resides in constructing case-specific components of the email from the input job and candidate to enhance personalization and persuasiveness. Quite often, the reasons of a good match are semantically non-explicit and cannot be extracted by common word or phrase matching. At the same time, we have observed that a simple approach can remarkably benefit recruiters by saving around 1/4 of their time.

Future work could contribute to this task in the following aspects:

- End-to-end neural generation of motivational sentences. Rather than the simple rule-based algorithm for composing the motivational sentences, we may learn a neural generation system from the real emails in a data-driven manner.

- Robust intrinsic evaluation metrics. The n-gram matching-based metrics, such as BLEU or ROUGE, are not suitable for this task. Beyond the real-user evaluation, it remains an open challenge to propose robust metrics that objectively evaluate the generated text, in terms of both language quality and task-related aspects.

- Deep representations of job and candidate to better extract reasons of match. Deep neural models should be constructed to perform deep understanding of jobs and candidates in order to support extraction of below-the-surface matched qualifications.

- Generating explanation of recommendations. Currently, the email generation is a separate process from job recommendation. It would be interesting to view recruitment emails instead as explanations of the recommendation systems, and propose recommendation models that supports interpretations of the recommendations. The authentic explanations of recommendation will be then be incorporated in the recruitment email.

Acknowledgements

This research was supported by the Innovation Fund Denmark, grant no. 0175-000005B.

References

Shahryar Baki, Rakesh Verma, Arjun Mukherjee, and Omprakash Gnawali. 2017. Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 469–482, New York, NY, USA. Association for Computing Machinery.

Ram Basnet, Srinivas Mukkamala, and Andrew H. Sung. 2008. *Detection of Phishing Attacks: A Machine Learning Approach*, pages 373–383. Springer Berlin Heidelberg, Berlin, Heidelberg.

Toine Bogers and Mesut Kaya. 2021. An Exploration of the Information Seeking Behavior of Recruiters. In *Proceedings of the RecSys in HR 2021 workshop*, pages 11–18.

Yun-Nung Chen and Alexander Rudnicky. 2014a. Two-stage stochastic email synthesizer. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 99–102, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Yun-Nung Chen and Alexander Rudnicky. 2014b. Two-stage stochastic natural language generation for email synthesis by modeling sender style and topic structure. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 152–156, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Avisha Das and Rakesh M. Verma. 2019. Automated email generation for targeted attacks using natural language. *CoRR*, abs/1908.06893.

Angela Fan and Claire Gardent. 2020. Multilingual AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2889–2901, Online. Association for Computational Linguistics.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Rakesh Verma and Ayman El Aassal. 2017. Comprehensive method for detecting phishing emailsusing correlation-based analysis and user participation. CODASPY '17, page 155–157, New York, NY, USA. Association for Computing Machinery.

Mingfeng Xue, Hang Zhang, and Jiancheng Lv. 2020. Key factors of email subject generation. In *Neural Information Processing*, pages 668–675, Cham. Springer International Publishing.

Weider D. Yu, Shruti Nargundkar, and Nagapriya Tiruthani. 2009. Phishcatch - a phishing detection tool. In *Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference - Volume 02*, COMPSAC '09, page 451–456, USA. IEEE Computer Society.

Rui Zhang and Joel R. Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. *CoRR*, abs/1906.03497.