Computational Job Market Analysis
with Natural Language Processing
Background
What is “Computational Job Market Analysis”?  
Our definition.

“The use of computational methods and tools to analyze trends, patterns, and dynamics within the job market.”
What is “Computational Job Market Analysis”?  

Our definition.

“The use of computational methods and tools to analyze trends, patterns, and dynamics within the job market.”

We leverage data science, machine learning, natural language processing, and other computational methods to gather, process, and interpret data related to job postings, employment statistics, labor market dynamics, skills demand, and other relevant factors.
What is “Computational Job Market Analysis”? Our definition.

“The use of computational methods and tools to analyze trends, patterns, and dynamics within the job market.”

We leverage data science, machine learning, natural language processing, and other computational methods to gather, process, and interpret data related to job postings, employment statistics, labor market dynamics, skills demand, and other relevant factors.
What are skills? (1)

Social scientist perspective.

• The concept of competence takes on diverse meanings across numerous fields and research domains (Rauner and Maclean, 2008, Zhao and Rauner, 2014):

  • A competence is having the necessary ability, authority, skill and knowledge. It is an individual’s psychological character;

  • For competence to be defined, many factors play an important role, e.g., education systems, cultures, economical structures, vocational traditions as well as situations in the labour market.
What are skills? (2)

A taxonomical perspective.

• The European Skills/Competences, Qualifications & Occupations taxonomy (le Vrang et al., 2014) —> first public release in 2020
  • 13,890 concepts (knowledge, language skills, skills, transversal skills)
  • 3,008 occupations
  • 28 languages
What are skills? (3)
A taxonomical perspective.

13,890 concepts (knowledge, language skills, skills, transversal skills)

• Two important “pillars”:

• **Knowledge**: “knowledge means the outcome of the assimilation of information through learning” ≈ *Hard skills*

• **Skill**: “the ability to apply knowledge and use know-how to complete tasks or solve problems” ≈ *Soft skills*

• + soft, transversal, and language skills
ESCO Illustrative Example.

The European Skills, Competences, Qualifications and Occupations Taxonomy (ESCO)

ESCO level 1: major group

S4 - Management skills
Example

ESCO Illustrative Example.

The European Skills, Competences, Qualifications and Occupations Taxonomy (ESCO)

ESCO level 1: major group

ESCO level 2: sub-major group

S4 - Management skills

S4.8 - Supervising people
The European Skills, Competences, Qualifications and Occupations Taxonomy (ESCO)

ESCO level 1: major group

ESCO level 2: sub-major group

ESCO level 3: minor group

S4 - Management skills

S4.8 - Supervising people

S4.8.1 - Supervising a team or group
Example
ESCO Illustrative Example.

The European Skills, Competences, Qualifications and Occupations Taxonomy (ESCO)

ESCO level 1: major group
S4 - Management skills

ESCO level 2: sub-major group
S4.8 - Supervising people

ESCO level 3: minor group
S4.8.1 - Supervising a team or group

ESCO level 4: unit group
S4.8.1.5 - Discharge employees
Example

ESCO Illustrative Example.

The European Skills, Competences, Qualifications and Occupations Taxonomy (ESCO)

ESCO level 1: major group

ESCO level 2: sub-major group

ESCO level 3: minor group

ESCO level 4: unit group

ESCO skills to occupation

S4 - Management skills

S4.8 - Supervising people

S4.8.1 - Supervising a team or group

S4.8.1.5 - Discharge employees

1212.2 - Human resources manager
Skill Extraction == Sequence Labeling.

You • General LM Encoder • handle • some • Python •

Output Layer

You • can • handle • some • Python •

B-Skill
Setting the NLP context (1)

Skill Extraction == Sequence Labeling.

You → General LM Encoder → Output Layer
  can →                          → 0
  handle →                      → 0
  some →                        → 0
  Py →                          → 0
  #thon →                      → B-Skill
  . →                          → I-Skill
  → 0
Setting the NLP context (2)

Skill Extraction == Sequence Labeling

You have critical thinking skills.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>General LM Encoder</td>
<td></td>
</tr>
<tr>
<td>You</td>
<td>O</td>
</tr>
<tr>
<td>have</td>
<td>O</td>
</tr>
<tr>
<td>critical</td>
<td>B-Skill</td>
</tr>
<tr>
<td>thinking</td>
<td>I-Skill</td>
</tr>
<tr>
<td>skills</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>0</td>
</tr>
</tbody>
</table>
Challenges
Challenges

Five Challenges with Skills.

1. **Implicitness**: The skill “teamwork” may manifest in various formats such as “being able to work together,” “working in a team,” or “collaborating efficiently.”

2. **Span Length**: Skills, by their nature, can exhibit considerable length, posing a challenge in extraction with NLP models.

3. **Long-tail**: Certain skills are infrequently mentioned in a comprehensive set of job postings.

4. **Discontinuity**: There is a discontinuity of skill spans, analogous to what is referred to as *coordinated conjunctions* in linguistics.

5. **Output Space**: If we use a taxonomy like ESCO as the labels, this results in a substantial output space during inference, adding a layer of complexity to the modeling process.
Challenges

Five Challenges with Skills.

1. **Implicitness**: The skill “teamwork” may manifest in various formats such as “being able to work together,” “working in a team,” or “collaborating efficiently.”

2. **Span Length**: Skills, by their nature, can exhibit considerable length, posing a challenge in extraction with NLP models.

3. **Long-tail**: Certain skills are infrequently mentioned in a comprehensive set of job postings.

4. **Discontinuity**: There is a discontinuity of skill spans, analogous to what is referred to as coordinated conjunctions in linguistics.

5. **Output Space**: If we use a taxonomy like ESCO as the labels, this results in a substantial output space during inference, adding a layer of complexity to the modeling process.
Computational Job Market Analysis Storyline

The red thread through the thesis.

Data
- JobStack (Jensen et al., 2021, NoDaLiDa)
- CAL (Zhang and Plank, 2021, EMNLP)
- SkillSpan (Zhang et al., 2022, NAACL)
- Kompetencer (Zhang et al., 2022, LREC)

Modeling
- WeakSupervision (Zhang et al., 2022, RecSysHR)
- ESCOXLM-R (Zhang et al., 2023, ACL)
- NNOSE (Zhang et al., 2024, EACL)

Linking
- EL Job Market (Zhang et al., 2024, EACL)
Computational Job Market Analysis Storyline

The red thread through the thesis.

**Data**
- JobStack (Jensen et al., 2021, NoDaLiDa)
- CAL (Zhang and Plank, 2021, EMNLP)
- SkillSpan (Zhang et al., 2022, NAACL)
- Kompetencer (Zhang et al., 2022, LREC)

**Modeling**
- WeakSupervision (Zhang et al., 2022, RecSysHR)
- ESOCXLM-R (Zhang et al., 2023, ACL)
- NNOSE (Zhang et al., 2024, EACL)

**Linking**
- EL Job Market (Zhang et al., 2024, EACL)
Getting Annotated Data
Getting Annotated Data

Why?

• We need annotated job posting data to train supervised models for skill extraction.

• Approaches:
  • Existing datasets (2 during this time);
  • Scraping + Annotating yourself (time-consuming).
Getting Annotated Data

Why?

• We need annotated job posting data to train supervised models for skill extraction.

• Approaches:
  • Existing datasets (2 during this time);
  • Scraping + Annotating yourself (time-consuming).
SkillSpan: Hard and Soft Skill Extraction from English Job Postings.

Mike Zhang, Kristian Nørgaard Jensen, Sif Dam Sonniks, and Barbara Plank.

Sources

Where does it come from?

• We have job postings from three sources:

  • **Tech**: Stackoverflow job postings;
  • **Big**: Indeed GB (not public);
  • **House**: Jobnet (thanks to STAR!)

![Styrelsen for Arbejdsmarked og Rekruttering](image)
Quick Recap ESCO

Knowledge and Skills

• 13,890 concepts (knowledge, language skills, skills, transversal skills)

• Two important “pillars”:

  • **Knowledge**: “knowledge means the outcome of the assimilation of information through learning” ≈ **Hard skills**

  • **Skill**: “the ability to apply knowledge and use know-how to complete tasks or solve problems” ≈ **Soft skills**

• + **soft, transversal, and language** skills
Example (1): Skills

SkillSpan: Hard and Soft Skill Extraction from English Job Postings.

“The successful candidate will be self-motivated and capable of working on their own initiative, an excellent communicator, with both customers and our highly motivated team at the company.”
Example (2): Soft-like skills

“We are looking for someone who is a team player with a reliable approach to problem solving.”
Example (3): Knowledge and nested components

SkillSpan: Hard and Soft Skill Extraction from English Job Postings.

“Experience in the kitchen industry and competent with computers. Strong experience in analyzing supply chain strategies.”
“Ability to read, write, configure, design, and script end-to-end service telemetry alerting and self-healing capabilities for platforms.”
Key Statistics

Dataset Statistics.

- 391 job postings:
  - 14.5K sentences;
    - 232K tokens;
  - 12.5K annotations;
    - 300 overlapping spans.
- 8 months of hard work.
Recap

Skill Extraction == Sequence Labeling

You • → General LM Encoder → Output Layer → 0

can • → 0

handle • → 0

some • → 0

Python • → B-Skill → 0

.
Two sets of experiments

SkillSpan

Domain-adaptive pretraining (Gururangan et al., 2020)

Multi-task Learning (Caruana, 1997)
Domain-adaptive Pre-training

Gururangan et al. (2020)
Domain-adaptive Pre-training

Gururangan et al. (2020)
Domain-adaptive Pre-training

Gururangan et al. (2020)

pre-training + domain-adaptive pre-training
+ fine-tuning
Single-task Learning

We can predict separately

Good • → LM Encoder → Output Layer → 0

Python • → LM Encoder → Output Layer → B-Knowledge → 0

Knowledge • → LM Encoder → Output Layer → B-Skill → I-Skill → 0

critical • → LM Encoder → Output Layer → B-Skill

critical • → LM Encoder → Output Layer → I-Skill

skills • → LM Encoder → Output Layer → B-Skill

skills • → LM Encoder → Output Layer → I-Skill

0

0
Multi-task Learning

Caruana (1997), hard parameter sharing
## Setup and Results

### Single Task vs. Multi Task Learning

- **Split of 200/90/101 JPs.**
- **Three models:**
  - BERT (Devlin et al., 2018)
  - JobBERT (ours, domain adapted)
  - JobSpanBERT (ours, domain adapted)
- **Single-task vs. multi-task learning (skills, knowledge, or combined)**
- **Single-task performs better than multi-task learning in most cases**

### Combined (Test)

<table>
<thead>
<tr>
<th>Model</th>
<th>Span F1</th>
<th>Single-task</th>
<th>Multi-task</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>58,16</td>
<td>57,73</td>
<td>57,73</td>
</tr>
<tr>
<td>JobBERT</td>
<td>59,73</td>
<td>59,18</td>
<td>59,18</td>
</tr>
<tr>
<td>JobSpanBERT</td>
<td>58,72</td>
<td>58,90</td>
<td>58,90</td>
</tr>
</tbody>
</table>

![Graph showing comparison between single-task and multi-task learning for BERT, JobBERT, and JobSpanBERT](image.png)
What does JobBERT work best on?

Challenge 2: Long spans

- JobBERT performs well on longer skills (3-7 tokens);
- Short skills include words such as “passionate” which could be a soft skill or not, could be challenging for the model;
What does JobBERT work best on?

Challenge 2: Long spans

- JobBERT performs well on longer skills (3-7 tokens);
- Short skills include words such as “passionate” which could be a soft skill or not, could be challenging for the model;
- Knowledge components usually consist of proper nouns (e.g., Python, Java) and might be easier for the model to predict.
Takeaways

• Domain-adaptive pre-training is a simple method to enhance your language model to a target domain.

• Knowledge components are easier to predict.

• Soft-like skills could pose a challenge to predict for language models.

• Challenge 2, Span Length: Skills, by their nature, can exhibit considerable length, posing a challenge in extraction with NLP models.

• On skills, our model predicts long sequences fairly well (3-7 tokens), but starts to degrade in performance with longer spans.
What about other languages?
What about other languages?

Apart from English

- Not everyone is searching for jobs in English;
- How can we make extracted spans useful for stakeholders?
  - **Classifying** the extracted spans to a standardized label.
  - In turn, can be used for counting skill demand.
Kompetencer: Fine-grained Skill Classification in Danish Job Postings via Distant Supervision and Transfer Learning

Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank.

In Proceedings of LREC 2022.
Similar to SkillSpan

But with extensions.

• Similar to SkillSpan, we annotated for skill and knowledge components in job postings.

• Two extensions:
  • Danish;
  • Classifying spans for their ESCO taxonomy code.
How did you get the annotations?

Challenge 5: Output Space.

• We rely on distant supervision (Mintz et al. 2009)

• Query ESCO API with the span to (hopefully) get a taxonomy code out.
  
  • In this work, we only consider the first major group (26 classes).

| Span 1 | ➔ | A1 (Attitudes) |
|———|———|———|
| Span 2 | ➔ | S4 (Management Skills) |
| Span 3 | ➔ | S2 (Information Skills) |
| Span 4 | ➔ | S7 (Constructing) |
| span 5 | ➔ | K01 (Education) |
Setting Context

Classifying Spans == Multiclass Classification (in this case).

“Individual work styles that can affect how well someone performs a job.”
Key Statistics

Dataset statistics.

- 60 job postings
- 1.5K sentences
- 20.3K tokens
- ~900 annotations
- 26 classes
- Data:
  - Skillspan (EN) and;
  - Kompetencer (DA)
Setup and Results

Monolingual vs. Multilingual Models

- **Split of 10 train (few-shot) / 50 test DA JPs.**
- **Zero-shot** (train on EN, apply to EN/DA)
Setup and Results

Monolingual vs. Multilingual Models

- **Split of 10 train (few-shot) / 50 test DA JPs.**
- **Zero-shot** (train on EN, apply to EN/DA)
- **Few-shot** (train on 10 DA JPs, apply to EN/DA)
Setup and Results

Monolingual vs. Multilingual Models

• Split of 10 train (few-shot) / 50 test DA JPs.

• Zero-shot (train on EN, apply to EN/DA), Few-shot (train on 10 DA JPs, apply to EN/DA)

• Five models:
  • EN: BERT (Devlin et al., 2018), JobBERT (Zhang et al., 2022)
  • DA: DaBERT (botxo), DaJobBERT (ours) (again domain-adaptive pre-training)
  • Multilingual: RemBERT (Chung et al., 2020)
Setup and Results

Monolingual vs. Multilingual Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Zero-shot Weighted Macro F1</th>
<th>Few-shot Weighted Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT (EN)</td>
<td>63.2</td>
<td>7.6</td>
</tr>
<tr>
<td>JobBERT (EN)</td>
<td>64.4</td>
<td>19.9</td>
</tr>
<tr>
<td>RemBERT (EN)</td>
<td>63.7</td>
<td>9.6</td>
</tr>
<tr>
<td>DaBERT (DA)</td>
<td>35.4</td>
<td>9.8</td>
</tr>
<tr>
<td>DaJobBERT (DA)</td>
<td>64.3</td>
<td>16.6</td>
</tr>
<tr>
<td>RemBERT (DA)</td>
<td>63.7</td>
<td>9.6</td>
</tr>
<tr>
<td>RemBERT (EN+DA)</td>
<td>63.2</td>
<td>7.6</td>
</tr>
</tbody>
</table>

Test (EN) | Test (DA)
Setup and Results

Monolingual vs. Multilingual Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Zero-shot</th>
<th>Few-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT (EN)</td>
<td>63.2</td>
<td>47.2</td>
</tr>
<tr>
<td>JobBERT (EN)</td>
<td>64.4</td>
<td>64.3</td>
</tr>
<tr>
<td>RemBERT (EN)</td>
<td>63.7</td>
<td>63.2</td>
</tr>
<tr>
<td>DaBERT (DA)</td>
<td>35.4</td>
<td>39.5</td>
</tr>
<tr>
<td>DaJobBERT (DA)</td>
<td>7.6</td>
<td>9.6</td>
</tr>
<tr>
<td>RemBERT (DA)</td>
<td>19.9</td>
<td>16.6</td>
</tr>
<tr>
<td>RemBERT (EN+DA)</td>
<td>35.4</td>
<td>7.6</td>
</tr>
</tbody>
</table>

Test (EN) and Test (DA) results.
## Setup and Results

### Monolingual vs. Multilingual Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Zero-shot</th>
<th>Few-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test (EN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT (EN)</td>
<td>63.2</td>
<td>64.4</td>
</tr>
<tr>
<td>JobBERT (EN)</td>
<td>64.4</td>
<td>63.7</td>
</tr>
<tr>
<td>RemBERT (EN)</td>
<td>63.7</td>
<td>35.4</td>
</tr>
<tr>
<td>DaBERT (DA)</td>
<td>7.6</td>
<td>19.9</td>
</tr>
<tr>
<td>DaJobBERT (DA)</td>
<td>9.6</td>
<td>39.5</td>
</tr>
<tr>
<td>RemBERT (DA)</td>
<td>9.6</td>
<td>16.6</td>
</tr>
<tr>
<td>RemBERT (EN+DA)</td>
<td>9.6</td>
<td>47.2</td>
</tr>
</tbody>
</table>

| Test (DA)              |           |          |
| DaBERT (DA)            | 63.2      | 64.4     |
| RemBERT (DA)           | 35.4      | 39.5     |
| RemBERT (EN+DA)        | 39.5      | 64.3     |
Setup and Results
Monolingual vs. Multilingual Models

- **Takeaway**: Even with little in-language data, adding it to your larger pool of training data with other languages can significantly improve in-language downstream performance.
Takeaways

Kompetencer

• Distantly supervised ESCO skill labels are a cost-effective approach to train models.
• We again show that domain-adaptive pre-training helps, beyond English.
• Larger multilingual models are better than smaller specialized monolingual models.
• **Challenge 5, Output Space**: If we use a taxonomy like ESCO as the labels, this results in a substantial output space during inference, adding a layer of complexity to the modeling process.
  • All models perform well for classifying skill spans with 26 classes. >64 W. Macro-F1 for EN and >47 W. Macro-F1 for Danish.
Tackling the Long Tail
Tackling the Long Tail

Label distribution problem.

• In job descriptions, there is a long-tail pattern, popular skills are more commonly mentioned, while niche expertise appears infrequently.

• This results in a sparsity of skills in skill extraction datasets.

• E.g., one dataset contains a single annotation for “identifying talent” (S4.1.1.14), but other datasets do not contain this, can a model generalize?

• Likely not.
NNOSE: Nearest Neighbor Occupational Skill Extraction

Mike Zhang, Rob van der Goot, Min-Yen Kan, and Barbara Plank.

To appear at EACL 2024.
A potential solution

NNOSE

• We explore Nearest Neighbor Language Models (NNLMs; Khandelwal et al., 2020), using the kNN algorithm as a retriever to retrieve context–token pairs from a datastore and leverage it during inference:
  • NNLMs have shown to memorize the training data better, aiding generalization;
  • NNLMs adapt to multiple domains without re-training;
  • NNLMs excels at predicting rare patterns, particularly in the long-tail.
• Concretely…
Overview of the idea

Fine-tuned LM Encoder

Aggregate Top-k

\[ p = \lambda p_{\text{kNN}} + (1 - \lambda)p_{\text{SE}} \]

Datastore

\( f: x \rightarrow x' \)

Query

Whitening Transformation

\( \{ f: K \rightarrow K', V \} \)

Token Tag

Knowledge of Python

\begin{array}{ll}
\text{B} & \text{I} \\
\text{B} & \text{I} \\
\text{B} & \text{O} \\
\end{array}

\begin{array}{ll}
\text{B} & \text{O} \\
\text{B} & \text{O} \\
\text{O} & \text{O} \\
\end{array}

\begin{array}{ll}
\text{B} & \text{O} \\
\text{B} & \text{O} \\
\text{O} & \text{O} \\
\end{array}

Representation

Predict

Fine-tuned LM Encoder
Setup

Models and Data

• Three models:
  • JobBERT (Zhang et al., 2022)
  • RoBERTa (Liu et al., 2019)
  • JobBERTa (This work), continuously pre-trained on 3.2M job posting sentences.

• Data:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Location</th>
<th>License</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>D (tokens)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SkillSpan</td>
<td>*</td>
<td>CC-BY-4.0</td>
<td>5,866</td>
<td>3,992</td>
<td>4,680</td>
<td>86.5K</td>
</tr>
<tr>
<td>Sayfullina</td>
<td>UK</td>
<td>UNK</td>
<td>3,706</td>
<td>1,854</td>
<td>1,853</td>
<td>53.1K</td>
</tr>
<tr>
<td>Green</td>
<td>UK</td>
<td>CC-BY-4.0</td>
<td>8,670</td>
<td>963</td>
<td>336</td>
<td>209.5K</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>348.2K</td>
</tr>
</tbody>
</table>

Table 1: Dataset Statistics. We provide statistics for all three datasets, including the location and license. Input granularity is at the token level, with performance measured in span-F1. The size of the datastore D is in tokens and determined by embedding tokens and their context from the training sets, resulting in approximately 350K keys.
Results

With and without kNN

- The best-performing baseline model is JobBERTa, achieving more than 4 points span-F1 improvement over JobBERT and 2 points higher than RoBERTa on average.
Results
With and without kNN

• The best-performing baseline model is JobBERTa, achieving more than 4 points span-F1 improvement over JobBERT and 2 points higher than RoBERTa on average.

• All models benefit from the NNOSE setup (+kNN), JobBERT and JobBERTa shows the largest improvements.
• We observe large gains in span-F1 using NNOSE in a cross-dataset setting.
• Trained on dataset X, applied to dataset Y.
• We confirm findings similar to Khandelwal et al. (2020), that memorisation using NNLMs improves recall.
(In)frequent Patterns, in-dataset

- Bin frequency of skills in the training set, ranging from 0–15 occurrences and is grouped into low, mid-low, mid-high, and high-frequency bins
  - (0–3, 4–6, 7–10, 10–15, respectively).
- Infrequent skills are the most difficult and make up the largest bucket, and our approach is able to improve on them on all three datasets.
Similar to the previous slide, we do the same analysis in the cross-dataset setting.

In the cross-dataset setting, we observe a large gain in performance when using NNOSE.
Takeaways

NNOSE

• Challenge 3, Long-tail: Certain skills are infrequently mentioned in a comprehensive set of job postings.
  • NNOSE performs well on rare skills and also enhances the performance of more frequent skill patterns.
  • NNOSE improves on all fronts in a cross-dataset setting.
• We show that NNOSE is a promising approach for application-specific skill extraction setups (e.g., Retrieval Augmentation).
Linking Skills to Existing Resources
Linking Skills to Existing Resources

How can we make the extracted skills useful?

• Previously, we have shown to be able to classify spans into their ESCO taxonomy code (26 classes).
  • Not taking surrounding context into account

• How can we scale this up to ~13,000 classes?
Entity Linking in the Job Market Domain

Mike Zhang, Rob van der Goot, and Barbara Plank.

To appear at Findings of EACL 2024.
Setting Context

Skill Matching == Entity Linking (Wu et al., 2020; Discriminative)

You can handle some Python.

Bi-Encoder

Description

... (~13k+)
Advising
Nursing
→ Python (Computer Programming)
3D Design
Forklifting
... (~13k)
Setting Context

Generative (de Cao et al. 2021)

You can handle some Python.

Encoder-Decoder

Python (Computer Programming)
Setting Context

Generative, GIF from de Cao (2021).

Who created the World Wide Web?
Setup

Two models and a benchmark

- Training data from Decorte et al. (2023), a synthetic training set generated using GPT-4.
- Dev. and Test are human-curated, Decorte et al. (2022) built on top of Zhang et al. (2022) by manually matching skill spans to the ESCO taxonomy.
- Models, two entity-linkers:
  - BLINK (Wu et al., 2020), discriminative
  - GENRE (Cao et al., 2021), generative

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Instances</th>
<th>Unique Titles</th>
<th>UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>123,619</td>
<td>12,984</td>
<td>14,641</td>
</tr>
<tr>
<td>Dev.</td>
<td>480</td>
<td>149</td>
<td>233</td>
</tr>
<tr>
<td>Test</td>
<td>1,824</td>
<td>455</td>
<td>813</td>
</tr>
</tbody>
</table>
We show that the discriminative linker (i.e., BLINK) is more suited for this task.

Strict Linking evaluation favours BLINK, however if we relax the number of candidates (i.e., higher # of k), we observe that GENRE performs better.
## Qualitative Analysis

### Challenge 1: What does it link?

<table>
<thead>
<tr>
<th>Mention</th>
<th>BLINK</th>
<th>GENRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Work in a way that is patient-centred and inclusive.</td>
<td>person centred care (K0913)</td>
<td>work in an organised manner (T)</td>
</tr>
<tr>
<td>(2) You can ride a bike.</td>
<td>sell bicycles (S1.6.1)</td>
<td>drive two-wheeled vehicles (S8.2.2)</td>
</tr>
<tr>
<td>(3) It is expected that you are a super user of the MS office tools.</td>
<td>use Microsoft Office (S5.6.1)</td>
<td>tools for software configuration management (0613)</td>
</tr>
<tr>
<td>(4) Picking and packing.</td>
<td>carry out specialised packing for customers (S6.1.3)</td>
<td>perform loading and unloading operations (S6.2.1)</td>
</tr>
<tr>
<td>(5) You are expected to be able to further develop your team - both personally and professionally.</td>
<td>manage personal professional development (S1.14.1)</td>
<td>shape organisational teams based on competencies (S4.6.0)</td>
</tr>
<tr>
<td><strong>GOLD: manage a team (S4.8.1)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6) Our games are developed using Unity so we expect all our programmers to have solid knowledge of mobile game development in Unity3D and C#.</td>
<td>C# (K0613)</td>
<td>C# (K0613)</td>
</tr>
</tbody>
</table>
Takeaways

EL in Job Market

• We show that the discriminative linker (i.e., BLINK) is more suited for this task.
• Strict Linking evaluation favours BLINK, however if we relax the number of candidates (i.e., higher # of k), we observe that GENRE performs better.
• Qualitative analysis shows that both models link valid predictions for a skill mention.
  • However, entity linking does not allow for multiple correct predictions, highlighting the need for a more comprehensive evaluation.

• Challenge 1: Entity Linking models have shown to be able to link implicit skills.
• Challenge 5: There is a reasonable performance linking skill spans to ~13k labels.
Conclusions
Conclusions

What can we learn from this?

• Skill Extraction is a hard task with many interesting challenges that could be of interest for general NLP research:
  • Long spans; Discontinuity; Implicitness; Long-tail extraction; Large label space.
• We show four contributions tackling four challenges in Skill Extraction.
  • CH1: Entity Linking models have shown to be able to link implicit skills;
  • CH2: Language models are able to predict long skills on a reasonable level (up to 7 tokens);
  • CH3: Retrieval-augmentation helps predicting skills in the long tail;
  • CH5: Models are able to match skills to their taxonomy counterpart, but need re-ranking methods to perform better.
Future Directions

Where do we go?

- **Beyond Job Postings**
  - Other type of document (e.g., resumes).

- **Large-scale End-to-End datasets**
  - Multiple span-level annotations of skills and linking them directly to a taxonomy code.
  - Synthetic data?

- **Large Language Models**
  - Conjoined skills can likely be split up by LLMs.
  - Constraining output space.
Thanks for listening!

Questions?
References


