SkillSence

Understanding skills beyond the surface

Yonatan Elman, Michael Kovalchuk, Roni Fadlon

Driving Use Case: Hidden Skills in Technical Resumes

The Problem

Tech recruitment relies on keyword-based ATS filtering, often missing candidates whose skills are expressed implicitly rather than as explicit tool names.

Importance

High-quality candidates are disqualified because they do not explicitly list specific tools or technologies.

The Challenge

High variability in phrasing and vague context make rule-based extraction less effective.

Solution Gap

Manual review is not scalable, and standard ATS tools miss implicit or vague evidence of skills.

Task Definition & Innovation

Goal

Detect, for a given technical resume, how each skill in a global skill list (~120–150 technical skills derived from 10 common tech job roles) is expressed in the CV text as explicit, implicit, or ambiguous.

Novelty

Explicit modeling of explicit vs. implicit vs. ambiguous skill evidence in full free-text CVs, over a predefined multi-role skill ontology, using controlled synthetic data.

Input

One full technical resume (CV) containing multiple job roles and their free-text descriptions

Output

For each skill in the global predefined skill list (~120–150 skills), output a (skill, label) pair, where label ∈ {NONE, EXPLICIT, IMPLICIT, AMBIGUOUS}.

Example:

"Designed and optimized SQL queries for dashboards" →(SQL, EXPLICIT); (Data analysis, IMPLICIT)

Skill Extraction Pipeline: Architecture and Models

1

Synthetic Data Generation

Leverage LLM-generated job descriptions with carefully controlled skill labels (NONE, EXPLICIT, IMPLICIT, AMBIGUOUS) over a global skill list (~120–150 skills derived from 10 tech job roles), ensuring diverse training data.

2

Fine-Tuned Transformer

Fine-tune advanced Transformer models (BERT/RoBERTa) to accurately map a single job description to a comprehensive label vector across a predefined skill set.

3

Skill Vector Prediction

Predict a 4-way label for each skill in the global skill list — NONE, EXPLICIT, IMPLICIT, or AMBIGUOUS — producing a skill-expression vector for every job description.

4

Full CV Aggregation

Aggregate job-level label vectors from all positions to construct a holistic technical skill profile for each CV, providing a complete candidate overview.

Main Model: Fine-tuned Transformer

Utilizes BERT/RoBERTa, taking a single job description as input and outputting a detailed label vector across all defined skills. This forms the core of our extraction system.

Baseline 1: Keyword Matching

Employs simple keyword rules based on skill names and common phrases. While straightforward, it is expected to have limitations in identifying implicit skills.

Baseline 2: Zero-Shot LLM

A more robust baseline that performs prompt-based skill extraction without additional training. This provides a strong comparative benchmark for model performance.

Refinement & Optimization

Focus on hyperparameter tuning and exploring prompt variants to significantly improve the model's ability to distinguish effectively between EXPLICIT, IMPLICIT, and AMBIGUOUS skill mentions.

Data Strategy: Controlled Synthetic Generation

- Q: What are the training and evaluation data requirements?

  A: Each example is a single job description text plus a label vector over a fixed global list of ~120–150 technical skills (derived from 10 tech job roles), with separate synthetic Train / Validation / Test splits.

- Q: What dataset will we use?

  A: Our main dataset is a large synthetic dataset of LLM-generated job descriptions that satisfy this format (job text + skill label vector over the global skill list).

- Q: What labeling are we going to perform (if any)?

  A: We sample a subset of skills from the global list, assign each sampled skill a label (EXPLICIT / IMPLICIT / AMBIGUOUS, all other skills = NONE), prompt an LLM to generate a matching job description, and automatically validate and store each example as (job_text, label_vector).

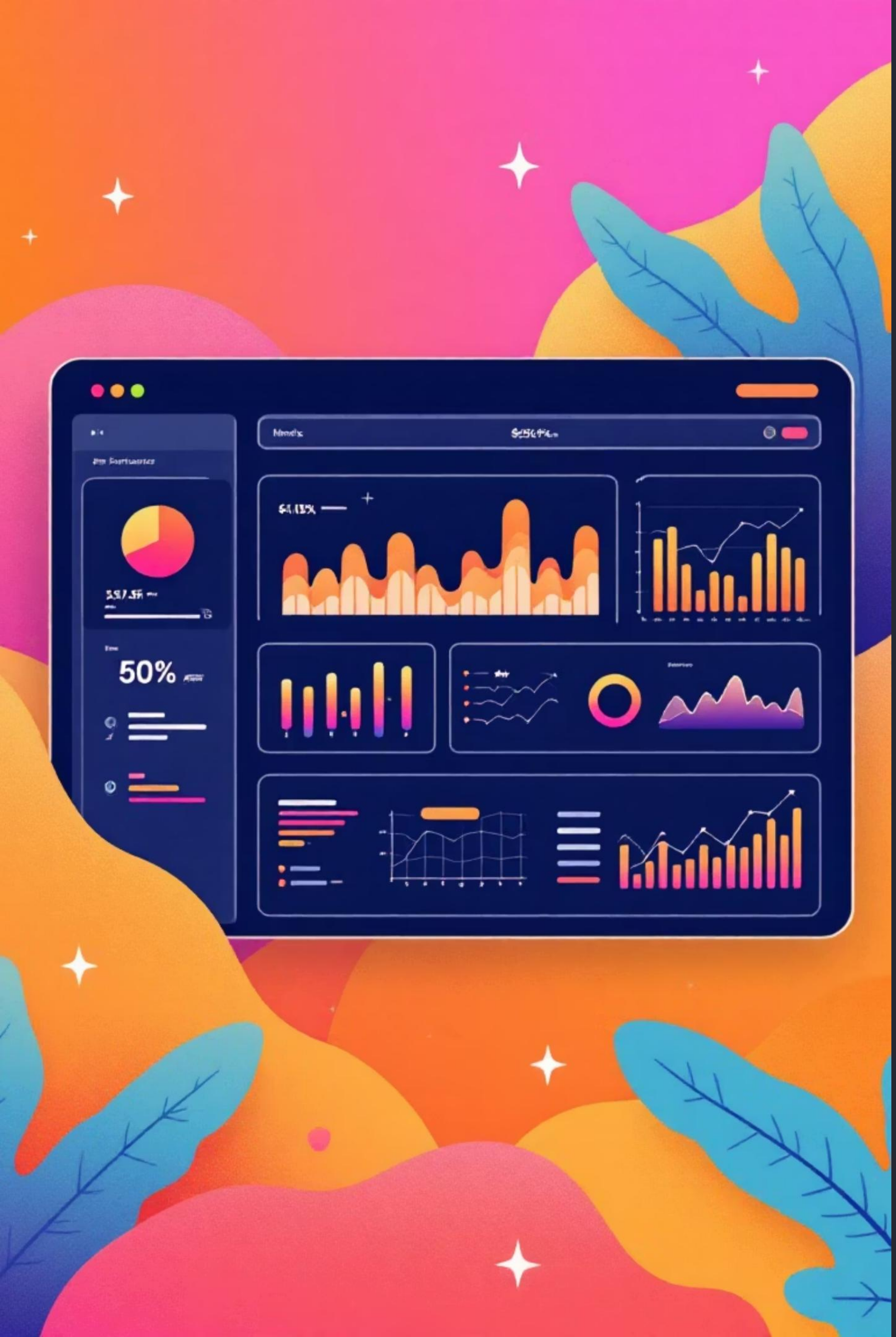
- Q: What synthetic data are we going to generate, and how?

  A: We sample a subset of skills from the global list, assign each sampled skill a label (EXPLICIT / IMPLICIT / AMBIGUOUS, all other skills = NONE), prompt an LLM to generate a matching job description, and automatically validate and store each example as (job_text, label_vector).

Data Strategy: Example

```
{
  "job_id": 12345,
  "job_text": "Designed dashboards in Tableau, collaborated with data engineers to define KPIs, and helped migrate services to containerized environments in the cloud.",
  "global_skills_vector": ["SQL", "Python", "Tableau", "Docker", "Kubernetes", "Data analysis"],
  "expression_type_vector": [2, 0, 1, 3, 0, 2]
}
```

- *global_skills_vector – unified global skill list aggregated from all 10 tech job roles.*
- *expression_type_vector – for each skill, an integer label describing how the skill is expressed in the job text (0–3 according to the mapping above).*
- *SQL → 2 (IMPLICIT) – not mentioned by name, but dashboards + KPIs + collaboration with data engineers strongly imply SQL work.*
- *Python → 0 (NONE) – no explicit or implicit evidence for Python.*
- *Tableau → 1 (EXPLICIT) – directly mentioned in "dashboards in Tableau".*
- *Docker → 3 (AMBIGUOUS) – "containerized environments in the cloud" suggests containers, but the specific tool is unclear (could be Docker, Kubernetes, etc.).*
- *Kubernetes → 0 (NONE) – not mentioned or clearly implied.*
- *Data analysis → 2 (IMPLICIT) – designing dashboards and KPIs implies data analysis, but the phrase "data analysis" is not used explicitly.*



Evaluation and Metrics

Evaluation Protocol

We evaluate how often the model predicts, for each (job, skill) pair, the correct 4-way label: NONE / EXPLICIT / IMPLICIT / AMBIGUOUS.
A correct prediction is, for example, "Python → IMPLICIT".

Success Criteria

Success = clearly outperforming the baselines in predicting the correct (skill, label) pairs – especially for IMPLICIT and AMBIGUOUS skills – while maintaining strong performance on EXPLICIT skills.

Evaluation Protocol

Train / validation / test splits on the synthetic corpus, evaluating all (job, skill) pairs against their predefined labels (NONE / EXPLICIT / IMPLICIT / AMBIGUOUS).

Key Metrics

Accuracy, Precision, Recall, and F1 per skill and label, with macro/micro averages and special focus on implicit labels.

Confusion Matrix

4-way confusion matrix over NONE / EXPLICIT / IMPLICIT / AMBIGUOUS, highlighting typical errors between IMPLICIT and AMBIGUOUS, and between NONE and IMPLICIT.

Success Criteria

Higher macro F1 and higher IMPLICIT / AMBIGUOUS F1 than both keyword and zero-shot LLM baselines, while maintaining strong F1 for EXPLICIT skills.