

Expansion Methods for Job-Candidate Matching Amidst Unreliable and Sparse Data

Jerome WHITE Krishna KUMMAMURU Nitendra RAJPUT

IBM Research, India

{jerome.white,kummamuru,rnitendra}@in.ibm.com

ABSTRACT

We address the problem of matching jobs with workers when information about both elements is incomplete and in some cases inaccurate. Such a situation occurs, for example, when profile information is generated from recorded audio, rather than typed or written sources. We present various methods of dealing with such post-processed voice information and show how it compares to human generated matches over the same data. Our analysis includes both SQL- and ontological-based methods that provide higher recall over a sparse data. A probabilistic weighted ontology model is proposed that enables assignment of realistic weights to different attributes and considers probabilistic conversion of audio to text. The evaluation is performed on real-life data from 1,100 candidates and 48 jobs spanning more than 3,000 vacancies.

KEYWORDS: Spoken Web, job search, resume matching, ontology.

1 Introduction

While job websites such as Dice and Monster have proven useful for a variety of job seekers, such technologies have not been suitable for unorganized, low-skilled workers in developing countries. To take full advantage of websites, users are required to have reliable Internet access, and some amount of technological savvy. To bring the benefits of the Internet to a larger population we have developed a *VoiceSite* (see [Kumar et al., 2007a](#)) to enable users to create and advertise their resumes through voice. Candidates and employers can listen to VoiceSites through telephone—including low-end mobiles that are prevalent in rural communities—in local languages. One of the challenges in developing such a system lies in data capture: being that we are dealing with voice audio, data is often noisy and not well structured. This paper presents various methods of generating “good” matches between candidates and employers amidst such unreliable information.

The difficulty of matching in this context is two-fold: first, attribute similarity is domain specific, not necessarily captured by traditional string or numeric distances; second, elements may have a subset of attributes defined due to the problems around voice-based data capture. Specifically, given that underlying values are taken from voice-data, they are both imprecise and lie within a wide range; resulting in a very sparse data set where exact matches are rare. Thus, to increase user satisfaction, we must look beyond exact matches.

While our results were developed for, and subsequently tailored to, low-skilled job matching, the methodologies discussed herein are applicable within the broader context of voice-driven market places. That is, environments where a supply must be matched to a demand, and the specifics of both are drawn from voice records. In many rural regions of developing countries, this is becoming a popular method of information exchange given the rate of mobile penetration versus the rate of Internet and literacy growth.

2 Related work

Recommender systems have been used to tackle pre-selection of candidates ([Malinowski et al., 2008](#)), as well as improve job-candidate match rankings ([Keim, 2007](#)). Automated methods have also been used to ease the burden on human decision making ([Yu et al., 2005a](#)). One novel approach has been to generate filters based on the information that is available in resumes ([Singh et al., 2010](#)). While most of these studies are performed either on synthetic data or on text data, our work is performed on real-world, semi-structured audio data.

Much of the work on spoken document retrieval (SDR) has been done on English language broadband speech (16–24 KHz audio) ([Singhal and Pereira, 1999](#); [Mamou et al., 2006](#); [Garofolo et al., 1999](#)). Speech recognition accuracies for such languages are usually more than 60 percent. However, this is not directly applicable to resource constrained languages and telephony speech (8 KHz audio), which is the domain of our work.

Over the last five years, researchers have started to look at searching speech not through speech recognition, but by indexing speech at a subword level ([Yu et al., 2005b](#); [Chelba and Acero, 2005](#)). Such techniques are more robust to speech recognition errors since they consider multiple recognition hypotheses in the index ([Chia et al., 2008](#)). Most systems for audio search use a visual query interface for accessing content indexed from audio. This is typical of a web based search system for searching video content ([Alberti et al., 2009](#)), and in call centers where supervisors wish to monitor offline content of the call center agents ([Mishne et al., 2005](#)). In such situations, the query is still textual. What differentiates our work is that both the query

ID	City	Qualification	Skill	Salary	Experience
669	Mandya	Diploma		5000	5
670	Mandya	Diploma		10000	5
681	Mandya	ITI	Mechanical	6000	4
684	Bangalore*	PUC	COPA	5500400	0
685	Kodagu*	PUC		6000	0
695	Bangalore*	BDes		6000	2
711	Bijapur*	MTech	Technician*	15000	2
712	Bijapur	PUC		8000	1
718	Bangalore	PUC	Data Entry Operator	7000	1
723	Bangalore	BCom		5000	5

Table 1: Job profiles used for matching. “Experience” is denoted in years; salary values are in denominations of Indian Rupees per month. The salary specification for job 684 is presumably a mistake in user input. Starred entries (*) are values obtained from offline speech recognition. Missing values are pieces of information that could be recognised neither during the call, nor during offline processing.

and the content are in audio.

Ontology based information extraction systems use semantic information to enrich the existing content for improving the understanding of the system (Chandrasekaran et al., 1999). Such systems are being increasingly used for various information extraction tasks: understanding the context (Cimiano et al., 2005), extracting content from Wikipedia (Wu et al., 2008), next-page prediction (Mabroukeh and Ezeife, 2009), and business intelligence (Saggion et al., 2007), among others. This paper uses the information in ontologies with respect to location, skill and qualification to improve recall in job-candidate matching scenario.

3 System Setup

Candidates and employers create their resume and job profiles, respectively, by calling into an interactive voice platform driven by Spoken Web (Kumar et al., 2007b). When a party calls into the system, they are presented with a series of prompts to collect their information. Addressing these prompts is done either by dual-tone multi-frequency signaling (DTMF), or by speaking an answer. For example, “please speak your highest qualification,” versus “using your phone keypad, please enter the monthly salary you are seeking.” The audio content is stored as 8 KHz data, and is in a mix of English and Kannada, the language native to the area in which the system was deployed.

There are five structured fields common to both candidates and employers: location, qualification (degree), skill, salary, and experience. Location, qualification, and skill were spoken input; salary and experience were DTME. Structured fields are converted to text and stored in a database. In the case of spoken fields, audio is first run through an online speech recognizer to perform the conversion. In the event that the recognizer cannot confidently produce a single answer, a list of probable answers is maintained, and the recorded audio is stored.¹

¹The list of probable answers is stored in a location that is separate from confident answers. Thus, database analysis can easily distinguish between the two.

3.1 Job Selection

Our analysis was performed over a subset of 10 jobs; [Table 1](#) provides details of each. These jobs were randomly selected from a subset of the 48 jobs that had similar candidate-distance distributions. Selected jobs fell into one of three categories: jobs with complete information (681, 718); jobs with incomplete but speech augmented information (684, 711); and jobs with information that have at least one attribute is incomplete (669, 670, 685, 695, 712, 723)—where speech recognition could not determine one or more of their attributes. This subset provided volunteers with room for personal subjectivity, and enough choices within the candidate pool that such subjectivity could be realized.

3.2 Ground Truth

The ground truth was established by asking a set of volunteers to manually match candidates to jobs. Ten volunteers were presented with three job openings. For redundancy, job was assigned to three separate volunteers; thus, each job had three sets of independently matched candidates.² All volunteers were presented with the same set of available candidates. Where candidate information was missing, textual results from the speech recognizer, along with recorded audio from candidates, were presented. It was clear to the volunteer, from presentation of the candidate database, where such information was included in place of standard database information. For example, if a candidates location was missing, the value would be empty; however, another value within the database would be filled with the recognized text and the recorded speech. Further, volunteers were made aware of the nature of this information—that it was inexact and perhaps not clear—and instructed to use their best judgment.

4 Methodology

We compare two job matching algorithms with a set of ground truth. Throughout, we denote the set of attributes as A , the set of jobs as J , and the set of candidates as C . It is sometimes convenient to talk about jobs and candidates as generic elements from the same set; in such cases $E = J \cup S$.

4.1 SQL Queries

Sets of job matches were generated directly from the database using structured query language (SQL). Two separate queries were performed: one that operated over DTMF and high confidence speech recognition values, and another operating over DTMF and probable speech recognition values. In the case of the former, if a particular attribute had a missing value, that attribute was not taken into account when performing the query. In the case of the latter, the highest probable recognized value was taken in place of missing values.³

In both cases, results were ranked by the number of attributes containing exact matches. For example, if there were n number of attributes to match, entries with n exact matches were ranked highest; matches with $n - 1$ entries were ranked second, and so forth. For entries with $n' < n$ exact attribute matches, a pre-defined priority was established for which n' subset was matched. Specifically, if two of three attributes matched, there were three different combinations of two-match pairings—a priority within that two-match set was specified. Based

²Volunteers were not only unaware of one another, but unaware that others might be investigating the same job.

³Ties were broken by selecting a value at random. If there was no value even in the probable set, then the attribute was not taken into account.

on prior research that involved interviews with the employers (White et al., 2012), qualification, location, and skill were prioritized in that order. As an example, an exact match of qualification and location will be ranked higher than an exact match of qualification and skill.

Our database was designed such that two independent tables contained job and candidate information. The `SELECT` and `FROM` portions of the SQL statement were straightforward, including the attributes of interest and the table of interest, respectively. A `WHERE` clause was used to combine pairs of attributes, while an `ORDER BY` clause was used to rank the results. Let $\mathcal{P}_{=2}(A)$ be the set of all attribute pairs, and $\mathcal{P}_{>1}(A)$ be the set of attributes where the cardinality is greater than one. The `WHERE` clause is the conjunction of $\mathcal{P}_{=2}(A)$ equalities,

$$\bigwedge_{(a_1, a_2) \in \mathcal{P}_{=2}(A)} (a_1 = v_e(a_1) \vee a_2 = v_e(a_2)),$$

where v_e is the value of an element $e \in E$ for a given attribute.

The `ORDER BY` clause consists of a case statement such that the more matching attributes that exist in a row, the higher the rows rank in the result set. Formally,

$$\forall A' \in \mathcal{P}_{>1}(A): \text{WHEN } \left[\bigwedge_{a \in A'} a = v_e(a) \right] \text{ THEN } i,$$

where $i = \max_{A' \in \mathcal{P}_{>1}(A)} |A'| - |A'|$. In practice, some augmentation is required as priority must be specified among subsets of $\mathcal{P}_{>1}(A)$ that have the same cardinality. Also, a final `ELSE`-clause must be specified with a value greater-than the cardinality of the largest set in $\mathcal{P}_{>1}(A)$.

As an example, consider a candidate whose location is Bangalore, skill is plumber, and qualification is PUC. The resulting SQL statement would be:

```
SELECT [columns] FROM [job_table] WHERE
  (location='Bangalore' OR skill='plumber') AND
  (skill='plumber' OR qualification='PUC') AND
  (qualification='PUC' OR location='Bangalore')
ORDER BY CASE
  WHEN location='Bangalore' AND skill='plumber' AND qualification='PUC' THEN 0
  WHEN location='Bangalore' AND skill='plumber' THEN 1
  WHEN location='Bangalore' AND qualification='PUC' THEN 2
  WHEN skill='plumber' AND qualification='PUC' THEN 3
  ELSE 4
END
```

4.2 Weighted Ontological Search

To broaden the scope of matches beyond exact string equality, we employed attribute-specific ontologies. These ontologies were applied to location, qualification, and skill. In particular, a notion of distance was developed for each pair of valid entries in each set. For location, the distance calculation corresponded to the Euclidean distance between two points. Distance between qualifications was developed based on the lattice that they formed. That is, when taken in order of academic completion—one must obtain a bachelors degree, for example, before obtaining a masters degree—qualifications form a tree. To determine a value for distance between qualifications, we used distance between their common parent.

Skills are a collection of arbitrary nouns. To establish distances between them we relied on WordNet (Miller, 1995). In particular, we used the methodology defined by Leacock et al. (1998) since, for our particular set of words, their algorithm provided highest variance among distances, with least number of zero values.

Let $j \in J$ and $c \in C$. Further, let w represent the weight of a given attribute for a particular element. The aggregate distance between a candidate and a job is thus the summation of the weighted distances between attributes,

$$D'(j, c) = \sum_{a \in A} w_a(j) \cdot d_a(j, c),$$

Attribute distances are normalized such that $d_a \in [0, 1]$ for all attributes a ; thus, $D \in [0, |A|]$. Note that comparisons that are exact matches, as in the case of SQL, will lie at the upper bound of this space ($D = |A|$).

For each attribute, weights were established by finding the variance of the distance between all elements. Formally, let $X_a(j) = \{d_a(j, c)\}_{c \in C}$, and $j \in J$,

$$\forall a \in A: w_a(j) = \begin{cases} 1 & \text{if } \text{Var}(X_a(j)) = 0, \\ \text{Var}(X_a(j))^{-1} & \text{otherwise.} \end{cases} \tag{1}$$

Equation 1 looks at the variance of a given attribute distances across all elements within a common set.⁴ It reduces the weight of a given attribute if there are several different distances for that attribute, giving precedence of attributes that are uniform. The logic being that searching between too many different people is difficult; by distributing that diversity among the common population, higher quality results arise.

5 Results

5.1 Evaluation

Because of the nature of our data, as well as the nature in which it is consumed, it is prudent to use a variety of retrieval metrics to establish the quality of our query methods. The inaccuracies in the data allow for a variety opinions when establishing matches; this was especially clear when evaluating the results within the ground truth itself. Thus, set-based metrics that do not consider rank have a place in our evaluation metrics, since there was no consensus even among human opinion. Further, in the context of job search, set inclusion is as important as rank. Because candidate search comes early in the recruitment stage, it is often a good idea for employers to select several candidates before beginning the initial screening process. Simply because a candidate is a good fit in theory does not mean they are a suitable fit in practice. Thus, an employer often wants to know a plurality of “best” matches, which make measures of unranked inclusion useful.

However, as previously mentioned, this data was acquired, and is meant to be consumed, via telephone. Such a medium not only has mental limitations, as audio data takes more patience to consume than visual data, but physical limitations as well. The longer a user waits to get results they are satisfied with, the longer they are tying up their line. This puts a strain on battery life, air-time charges, and ultimately user patience. Thus, comparing result rankings is also important.

⁴By “common” we mean candidates in the case of candidates, and jobs in the case of jobs

ID	N	SQL				Weighted Ontology			
		Prec@10	Rec@10	Avg Pr	ERR	Prec@10	Rec@10	Avg Pr	ERR
669	21	0.2500	0.0476	0.0159	0.0129	0.2000	0.0952	0.0417	0.0252
670	27	0.2500	0.0370	0.0093	0.0163	0.3000	0.1111	0.1069	0.0309
681	20	0.5000	0.1000	0.1833	0.0000	0.3000	0.0500	0.1500	0.0000
684	15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
685	34	0.0000	0.0000	0.0009	0.0000	1.0000	0.2941	0.6279	0.1202
695	25	0.0000	0.0000	0.0000	0.0000	0.4000	0.1600	0.1733	0.0884
711	6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
712	22	0.3000	0.1364	0.1297	0.0116	0.3000	0.1364	0.1297	0.0116
718	24	0.1000	0.0417	0.0469	0.0016	0.1000	0.0417	0.0117	0.0027
723	8	0.2857	0.2500	0.2083	0.0093	0.2500	0.2500	0.2083	0.0093
MAP				0.0594				0.1450	

Table 2: Retrieval metrics when searching over non-speech corrected data.

To capture these concepts, we use four metrics for evaluating our algorithms: recall, precision, average precision, and expected reciprocal rank (ERR) (Chapelle et al., 2009). Precision and recall are good measures for basic query similarity. Moreover, for documents retrieved via SQL, these metrics are in some ways a fairer perspective, as SQLs ability to rank is not as granular as the ontological-based methods.

Average precision and ERR are our ranked evaluation metrics. Again, given our context and environment, both metrics are relevant. On the one hand, when searching for candidates an employer is likely to go through a large set, choosing to continue their search irrespective of the goodness the current document provides. Of course, this is under the assumption that a majority of the returned set contains relevant documents. However, given that the employer is consuming these documents over the phone, they are likely to base some of their decision to continue on the quality of the current, or previous n , results, where n is small. We use the combination of average precision and ERR to capture these characteristics.

Note, that our use of metrics are primarily as a tool to compare our methodology. As such we are not necessarily concerned with their absolute value, but more so with their respective, or comparative, values. Further, when producing metrics, we used the number of results returned in the ground truth set as the number of documents we limited our searches to returning.

5.2 Evaluation Over Raw Data

We first apply our match heuristics to *raw data*. Raw data is data in which there is no attempt to correct the speech anomalies; thus, we ignore portions of the data for which there is no reliably accurate information. For example, if a candidate or employer spoke a particular attribute that our system did not recognize, we treat the data as missing. Comparing our results to the ground truth in this context was not entirely fair, as volunteers had access to spoken audio files and were not told to produce two sets of matches based on whether or not they listened to them. However, performing the comparison allows us to establish a baseline, and gives us an idea of how accurate the additional recognition information is.

Despite the lack of information, the ontology methods were able to outperform SQL queries

ID	N	SQL				Weighted Ontology			
		Prec@10	Rec@10	Avg Pr	ERR	Prec@10	Rec@10	Avg Pr	ERR
669	21	0.2500	0.0476	0.0159	0.0129	0.4000	0.1905	0.1317	0.0414
670	27	0.2500	0.0370	0.0093	0.0163	0.3000	0.0741	0.0966	0.0143
681	20	0.5000	0.1000	0.1833	0.0000	0.3000	0.0500	0.0300	0.0000
684	15	0.2500	0.0667	0.0222	0.0000	0.1000	0.0667	0.0095	0.0000
685	34	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0845	0.0152
695	25	1.0000	0.0800	0.0800	0.0435	0.3000	0.1200	0.1784	0.0605
711	6	1.0000	0.1667	0.1667	0.0000	0.1667	0.1667	0.1667	0.0000
712	22	0.3000	0.1364	0.1297	0.0116	0.3000	0.1364	0.1157	0.0097
718	24	0.1000	0.0417	0.0469	0.0016	0.2000	0.0883	0.0571	0.0290
723	8	0.2857	0.2500	0.2083	0.0093	0.2500	0.2500	0.1250	0.0319
MAP				0.0862				0.0995	

Table 3: Retrieval metrics when when searching over speech corrected data.

in rank evaluations for many of the job profiles; this trend is also see in recall and average precision. See [Table 2](#) for a comparison.

5.3 Evaluation with Noise Reduction

To reduce the amount of missing data in the system, we replaced non-existent values with estimates from the offline speech recognition engine. For each attribute, we first considered the value, or set of values, that the speech recognizer returned, taking the value with the highest probability as the authority. If, however, the recognition engine was unable to produce any values for a given attribute, we looked at recognition values of other attributes for that job or candidate. In some cases, we were actually able to find proper values in non-proper fields. As an example, consider a candidate that does not have a real-time recognized value for location. We would first check the offline recognition values for the recorded location. In the event that there were no values or valid values found, we would move to another speech attribute that was unrecognized, searching through that set for what could be considered a location. The premise was that some users might have had confusion during their usage of the system, giving valid input at incorrect times. These techniques for removing noise more than doubled the amount of overall information available for querying.

[Table 3](#) lists retrieval evaluation values for noise reduction. SQL performance was notably better, with respect to its own performance in the raw tests, as well as ontological methods within noise-reduced tests. In fact, recall measures aside, there is no clear winner between the two methods when offline speech recognition is present.

Conclusion and Future Work

Voice-based information processing, as tackled herein, is an important step in the success of mobile data collection. To this end, we have developed an algorithm to match jobs with candidates that considers both issues of noise in the data, as well as proximity of the attributes in matching. In the future, we would like to augment our matching function to consider all probable values presented by offline speech recognition, and to apply “active learning” to our weight calculation.

References

- Alberti, C., Bacchiani, M., Bezman, A., Chelba, C., Drofa, A., Liao, H., Moreno, P., Power, T., Sahuguet, A., Shugrina, M., and Siohan, O. (2009). An audio indexing system for election video material. In *International Conference on Acoustics, Speech, and Signal Processing*.
- Chandrasekaran, B., Josephson, J., and Banjamins, V. R. (1999). What are ontologies, and why do we need them? *IEEE Transactions on Intelligent Systems and their Applications*, 14(1).
- Chapelle, O., Metzler, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630.
- Chelba, C. and Acero, A. (2005). Position specific posterior lattices for indexing speech. In *Annual Meeting on Association for Computational Linguistics*, pages 443–450.
- Chia, T. K., Sim, K. C., Li, H., and Ng, H. T. (2008). A lattice-based approach to query-by-example spoken document retrieval. In *International Conference on Research and Development in Information Retrieval*, pages 363–370. ACM.
- Cimiano, P., Ladwig, G., and Staab, S. (2005). Gimme’ the context: context-driven automatic semantic annotation with c-pankow. In *international conference on World Wide Web*, pages 332–341, New York, NY, USA. ACM.
- Garofolo, J., Auzanne, C., and Voorhees, E. (1999). The trec spoken document retrieval track: A success story. In *TREC*.
- Keim, T. (2007). Extending the applicability of recommender systems: A multilayer framework for matching human resources. In *Annual Hawaii International Conference on System Sciences, HICSS ’07*, pages 169–, Washington, DC, USA. IEEE Computer Society.
- Kumar, A., Rajput, N., Chakraborty, D., Agarwal, S., and Nanavati, A. A. (2007a). Voiserv: Creation and delivery of converged services through voice for emerging economies. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Finland.
- Kumar, A., Rajput, N., Chakraborty, D., Agarwal, S., and Nanavati, A. A. (2007b). WWTW: A World Wide Telecom Web for Developing Regions. In *SIGCOMM Workshop on Networked Systems For Developing Regions*. ACM.
- Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics—Special issue on word sense disambiguation*, 24(1):147–165.
- MaBroukeh, N. and Ezeife, C. (2009). Using domain ontology for semantic web usage mining and next page prediction. In *Conference on Information and Knowledge Management*, pages 1677–1680, New York, NY, USA. ACM.
- Malinowski, J., Weitzel, T., and Keim, T. (2008). Decision support for team staffing: An automated relational recommendation approach. *Decision Support Systems*, 45(3):429 – 447.
- Mamou, J., Carmel, D., and Hoory, R. (2006). Spoken document retrieval from call-center conversations. In *International Conference on Research and Development in Information Retrieval*, pages 51–58, New York, NY, USA. ACM.

- Miller, G. (1995). WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Mishne, G., Carmel, D., Hoory, R., Roytman, A., and Soffer, A. (2005). Automatic analysis of call-center conversations. In *International conference on Information and knowledge management*, pages 453–459.
- Saggion, H., Funk, A., Maynard, D., and Bontcheva, K. (2007). Ontology-based information extraction for business intelligence. In *International Semantic Web Conference and Asian Semantic Web Conference*, pages 843–856, Berlin, Heidelberg. Springer-Verlag.
- Singh, A., Catherine, R., Visweswariah, K., Chenthamarakshan, V., and Kambhatla, N. (2010). PROSPECT: a system for screening candidates for recruitment. In Huang, J., Koudas, N., Jones, G. J. F., Wu, X., Collins-Thompson, K., and An, A., editors, *International Conference on Information and Knowledge Management*, pages 659–668. ACM.
- Singhal, A. and Pereira, F. (1999). Document expansion for speech retrieval. In *International Conference on Research and Development in Information Retrieval*, pages 34–41, New York, NY, USA. ACM.
- White, J., Duggirala, M., Kummamuru, K., and Srivastava, S. (2012). Designing a voice-based employment exchange for rural india. In *International Conference on Information and Communication Technologies and Development*, pages 367–373, New York, NY, USA. ACM.
- Wu, F., Hoffmann, R., and Weld, D. S. (2008). Information extraction from wikipedia: moving down the long tail. In *International conference on Knowledge discovery and data mining*, pages 731–739, New York, NY, USA. ACM.
- Yu, K., Guan, G., and Zhou, M. (2005a). Resume information extraction with cascaded hybrid model. In *Annual Meeting on Association for Computational Linguistics*, pages 499–506, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yu, K. C., Ma, C., and Seide, F. (2005b). Vocabulary independent indexing of spontaneous speech. *IEEE Transactions on Speech and Audio Processing*, 13(5).