Instance Sense Induction from Attribute Sets

Ricardo Martin-Brualla
Google Inc
rmbrualla@gmail.com

Enrique Alfonseca
Google Inc
ealfonseca@google.com

Marius Pasca
Google Inc
mars@google.com

Keith Hall
Google Inc
kbhall@google.com

Enrique Robledo-Arnuncio
Google Inc
era@google.com

Massimiliano Ciaramita
Google Inc
massi@google.com

Abstract

This paper investigates the new problem of automatic sense induction for instance names using automatically extracted attribute sets. Several clustering strategies and data sources are described and evaluated. We also discuss the drawbacks of the evaluation metrics commonly used in similar clustering tasks. The results show improvements in most metrics with respect to the baselines, especially for polysemous instances.

1 Introduction

Recent work on information extraction increasingly turns its attention to the automatic acquisition of open-domain information from large text collections (Etzioni et al., 2008). The acquired information typically includes instances (e.g. *barack obama* or *hillary clinton*), class labels (e.g. *politician* or *presidential candidate*) and relations and attributes of the instances (e.g. *president-country* or *date-of-birth*) (Sekine, 2006; Banko et al., 2007).

Within the larger area of relation extraction, the acquisition of instance attributes (e.g. *president* for instances of countries, or *side effects* for instances of drugs) plays an important role, since attributes may serve as building blocks in any knowledge base constructed around open-domain classes of instances. Thus, a variety of attribute extraction methods mine textual data sources ranging from unstructured (Tokunaga et al., 2005) or structured (Cafarella et al., 2008) text within Web documents, to human-compiled encyclopedia (Wu et al., 2008; Cui et al., 2009) and Web search query logs (Paşca and Van Durme, 2007), attempting to extract, for a given class, a ranked list of attributes that is as comprehensive and accurate as possible.

Previous work on attribute extraction, however, does not capture or address attributes of polysemous instances. An instance may have different meanings, and the extracted attributes may not apply to all of them. For example, the most salient meanings of *darwin* are the scientist *Charles Darwin*, an Australian city, and an operating system, plus many less-known meanings. For these ambiguous instances, it is common for the existing procedures to extract mixed lists of attributes that belong to incompatible meanings, e.g. {*biography, population, hotels, books*}.

This paper explores the problem of automatically inducing instance senses from the learned attribute lists, and describes several clustering solutions based on a variety of data sources. For that, it brings together research on attribute acquisition and on word sense induction. Results show that we can generate meaninful groupings of attributes for polysemous instance names, while not harming much the monosemous instance names by generating unwanted clusters for them. The results are much better than for a random baseline, and are superior to the one-in-all and the all-singleton baselines.

2 Previous Work

Previous work on **attribute extraction** uses a variety of types of textual data as sources for mining attributes. Some methods take advantage of structured and semi-structured text available within Web documents. Examples of this are the use of markup information in HTML documents to extract patterns and clues around attributes (Yoshinaga and Torisawa, 2007; Wong and Lam, 2009; Ravi and Paşca, 2008), or the use of articles within online encyclopedia as sources of structured text for attribute extraction (Suchanek et al., 2007; Nastase and Strube, 2008; Wu and Weld, 2008). Regarding unstructured text in Web documents, the method described in (Tokunaga et al., 2005) takes various class labels as input, and applies manually-created lexico-syntactic patterns to document sentences to extract candidate attributes ranked using several frequency statistics. In (Bellare et al., 2007), the extraction is guided by a set of seed instances and attributes rather than handcrafted patterns, with the purpose of generating training data and extract new instance-attribute pairs from text.

Web search queries have also been used as a data source for attribute extraction, using lexico-syntactic patterns (Paşca and Van Durme, 2007) or seed attributes (Paşca, 2007) to guide the extraction, and leading to attributes of higher accuracy than those extracted with equivalent techniques from Web documents (Paşca et al., 2007).

Another related area to this work is the field of **word sense induction**: the task of identifying the possible senses of a word in a corpus using unsupervised methods (Yarowsky, 1995), as opposed to traditional disambiguation methods which rely on the availability of a finite and static list of possible meanings. In (Agirre and Soroa, 2007) a framework is proposed for evaluating such systems. Word sense induction can be naturally formulated as a clustering task. This introduces the complication of choosing the right number of possible senses, hence a Bayesian approach to WSI was proposed which deals with this problem within a principled generative framework (Brody and Lapata, 2009). Another related line of work is the disambiguation of people names (Mann and Yarowsky, 2003). In SEMEVAL-1, a shared task was introduced dedicated to this problem, the Web People Search task (Artiles et al., 2007; Artiles et al., 2009). Disambiguating names is also often approached as a clustering problem. One challenge shared by word sense induction and name disambiguation (and most unsupervised settings), is the evaluation. In both tasks, simple baselines such as predicting one single cluster tend to outperform more sophisticated approaches (Agirre and Soroa, 2007; Artiles et al., 2007).

Turkey Attributes		**Darwin Attributes**	
maps$_1$	capital$_1$	maps$_1$	definition$_{1,3}$
recipes$_2$	culture$_1$	awards$_2$	jobs$_1$
pictures$_{1,2}$	history$_1$	shoes$_1$	tourism$_1$
calories$_2$	tourism$_1$	evolution$_3$	biography$_3$
facts$_{1,2}$	nutrition facts$_2$	theory$_3$	attractions$_1$
nutrition$_2$	beaches$_1$	weather$_1$	hotels$_1$
cooking time$_2$	brands$_2$	pictures$_{1,3}$	ports$_4$
religion$_1$	language$_1$	quotes$_3$	population$_1$

Table 1: Attributes extracted for the instances *Turkey* and *Darwin*.

3 Instance Sense Induction

3.1 Problem description

This paper assumes the existence of an attribute extraction procedure. Using those attributes, our aim is to identify the coarse-grained meanings with which each attribute is associated. As an example, Table 1 shows the top 16 attributes extracted using the procedure described in (Paşca and Van Durme, 2007). Salient meanings for *turkey* are the country name (labeled as 1 in the table), and the bird name (labeled as 2). Some attributes are applicable to both meanings (*pictures* and *facts*). The second example, *darwin*, can refer to a city (sense 1), the Darwin Awards (sense 2), the person (sense 3), and an operating system (sense 4).

Examples of applications that need to discriminate between the several meanings of instances are user-facing applications requiring the attributes to be organized logically and information extraction pipelines that depend on the extracted attributes to find values in documents.

The problem we are addressing is the automatic induction of instance senses from the attribute sets, by grouping together the attributes that can be applied to a particular sense. As in related work on sense induction (Agirre and Soroa, 2007; Artiles et al., 2007), we approach this as a clustering problem: finding the right similarity metrics and clustering procedures to identify sets of related attributes in an instance. We propose a clustering based on the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), exploring different parameters, similarity sources, and prior distributions.

3.2 Instance and attributes input data

The input data of instances and attributes has been obtained, in a fully automated way, following the method described in (Paşca and Van Durme, 2007). The input dataset is a set of fully anonymized set of English queries submitted to a popular (anonymized) search engine. The set contains millions of unique isolated, individual queries that are independent from one another. Each query is accompanied by its frequency of occurrence in the query logs. The sum of frequencies of all queries in the dataset is hundreds of millions. Other sources of similar data are available publicly for research purposes (Gao et al., 2007). This extraction method applies a few patterns (e.g., *the* $\mathcal{A}$ *of* $\mathcal{I}$, or $\mathcal{I}$*'s* $\mathcal{A}$, or $\mathcal{A}$ *of* $\mathcal{I}$) to queries within query logs, where an instance $\mathcal{I}$ is one of the most frequent 5 million queries from the repository of isolated queries, and $\mathcal{A}$ is a candidate attribute. For each instance, the method extracts ranked lists containing zero, one or more attributes, along with frequency-based scores. For this work, only the top 32 attributes of each instance were used, in order to have an input set for the clustering with a reasonable size, but to keep precision at high levels.

3.3 Per-attribute clustering information

For each (instance, attribute) pair, the following information is collected:

Search results: The top 20 search results (including titles and snippets) returned by a popular search engine for a query created by concatenating the instance and the attribute. The motivation for this data source is that the attributes that refer to the same meaning of the instance should help the search engine in selecting web pages that refer to that meaning. The titles and snippets of these search results are expected to contain other terms related to that meaning. For example, for the queries *[turkey maps]* and *[turkey culture]* the search results will contain information related to the country, whereas *[turkey recipes]* and *[turkey nutritional value]* should share many terms about the poultry.

Query sessions: A query session is a series of queries submitted by a single user within a small range of time (Silverstein et al., 1999). Information stored in the session logs may include the text of the queries and metadata, such as the time, the type of query (e.g., using the normal or the advance form), and user settings such as the Web browser used (Silverstein et al., 1999).

Users often search for related queries within a session: queries on the *culture* of the country Turkey will tend to be surrounded by queries about topics related to the country; similarly, queries about turkey *recipes* will tend to be surrounded by other queries on recipes. Therefore, if two attributes refer to the same meaning of the instance, the distributions of terms that co-occur with them in the same search sessions is expected to be similar. To ensure that the user did not change intent during the session, we also require the queries from which we extract phrases to contain the instance of interest. The pseudocode of the procedure is shown in Figure 1.

```
For each (instance, attribute) pair:
    • Retrieve all the sessions that contained the query [instance attribute].
    • Collect the set of all the queries that appeared in the same session and which are a superstring of instance.
    • Remove instance from each of those queries, and output the resulting set of query words.
```

Figure 1: Algorithm to collect session phrases associated to attributes.

Class labels: As described in (Paşca and Van Durme, 2008), we collect for each instance (e.g., *turkey*), a ranked list of class labels (e.g., *country, location, poultry, food*). The procedure uses a collection of Web documents and applies some IsA extraction patterns selected from (Hearst, 1992). Using the (instance, ranked-attributes) and the (instance, ranked-class labels) lists, it is possible to aggregate the two datasets to obtain, for each attribute, the class labels that are most strongly associated to it (Figure 2).

3.4 EM clustering

We run a set of EM clusterings separately for the attributes of each instance. The model implemented is the following: given an instance, let $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ be the set of attributes associated with that instance. Let $\mathcal{T}$ be the vocabulary for the terms found in the search results, $\mathcal{S}$ the vocabulary of session log terms co-occurring with

For each attribute:

- Collect all the instances that contain that attribute.
- For each class label, average its ranks for those instances. If an instance does not contain a particular class label, use as rank the size of the longest list of class labels plus one.
- Rank the class labels from smaller to larger average rank.

Figure 2: Algorithm to collect class labels associated to attributes.

the attribute, and $\mathcal{C}$ be the set of all the possible class labels. Let $\mathcal{K}$ be the cluster function which assigns cluster indexes to the attributes.

We assume that the distributions for snippet terms, session terms and class labels are conditionally independent given the clustering. Furthermore, we assume that the distribution of terms for queries in a cluster are also conditionally independent given the cluster assignments:

$$\begin{aligned} p_\theta(\mathcal{T}|\mathcal{K},\mathcal{A}) &\approx \prod_j p_\theta(t_j|\mathcal{K},\mathcal{A}) \\ p_\theta(\mathcal{S}|\mathcal{K},\mathcal{A}) &\approx \prod_k p_\theta(s_k|\mathcal{K},\mathcal{A}) \\ p_\theta(\mathcal{C}|\mathcal{K},\mathcal{A}) &\approx \prod_l p_\theta(c_l|\mathcal{K},\mathcal{A}) \end{aligned}$$

The clustering model for each instance (the expectation step) is, therefore:

$$p_\theta(\mathcal{KTSC}|\mathcal{A},\Theta) = \prod_i^N p_\theta(\mathcal{K}|\mathcal{A})p_\theta(\mathcal{T}|\mathcal{K},\mathcal{A})p_\theta(\mathcal{S}|\mathcal{K},\mathcal{A})p_\theta(\mathcal{C}|\mathcal{K},\mathcal{A})$$

To estimate the parameters of the model, we must be able to estimate the following distributions during the maximization step:

- $p_\theta(t_j|\mathcal{K},\mathcal{A}) = \frac{E_\theta(t_j,\mathcal{K}|\mathcal{A})}{E_\theta(\mathcal{K}|\mathcal{A})}$
- $p_\theta(s_k|\mathcal{K},\mathcal{A}) = \frac{E_\theta(s_k,\mathcal{K}|\mathcal{A})}{E_\theta(\mathcal{K}|\mathcal{A})}$
- $p_\theta(c_l|\mathcal{K},\mathcal{A}) = \frac{E_\theta(c_l,\mathcal{K}|\mathcal{A})}{E_\theta(\mathcal{K}|\mathcal{A})}$

One advantage of this approach is that it allows using a subset of the available data sources to evaluate their relative influence on the clustering quality. In the experiments we have tried all possible combinations of the three data sources to find the settings that give the best results.

3.5 Initialization strategies

The initial assignment of attributes to clusters is important, since a bad seed clustering can lead EM to local optima. We have tried the following two strategies:

Random assignment: the attributes are assigned to clusters randomly. To make the results repeatable, for each instance we use the instance name as the seed for the random number generator.

K-means: the initial assignments of attributes to clusters is performed using K-means. In this model, we use a simple vector-space-model in the following way:

1. Each attribute is represented with a bag-of-words of the snippets of the search results for a concatenation of the instance name and the attribute. This is the same data already collected for EM.
2. Each of the snippet terms in these bag-of-words is weighted using the $tf \times idf$ score, with inverse document frequencies estimated from an English web corpus with hundreds of millions of documents.
3. The cosine of the angle of the vectors is used as the similarity metric between each pair of attributes.

Several values of K have been tried in our experiments, as mentioned in Section 4.

3.6 Post-processing

EM works with a fixed set of clusters. In order to decide which is the optimal number of clusters, we have run all the experiments with a number of clusters K that is large enough to accommodate most of the queries in our dataset, and we run a post-processing step that merges clusters for instances that have less than K meanings.

Since we have, for each attribute, a distribution of the most likely class labels (Section 3.3), the post-processing performs as follows:

1. Generate a list of class labels per cluster, by combining the ranked lists of per-attribute class labels as was done in Section 3.3.
2. Merge together all the clusters such that their sets of top k class labels are the same.

The values of K and k are chosen by doing several runs with different values on the development set, as described in Section 4.

4 Evaluation and Results

4.1 Evaluation metrics

There does not exist a fully agreed evaluation metric for clustering tasks in NLP (Geiss, 2009; Amigó et al., 2009). Each metric has its own idiosyncrasies, so we have chosen to compute six different evaluation metrics as described in (Amigó et al., 2009). Empirical results show they are highly correlated, i.e., tuning a parameter by hill-climbing on F-score typically also improves the B^3 F-score.

Purity (Zhao and Karypis, 2002): Let C be the clusters to evaluate, L the set of categories (the clusters in the gold-standard), and N the number of clustered items. Purity is the average of the precision values: Purity $= \sum_i \frac{|C_i|}{N} \max_j \text{Prec}(C_i, L_j)$, where the precision for cluster C_i with respect to category L_j is $\text{Prec}(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$. Purity is a precision metric. Inverting the roles of the categories L and the clusters C gives a recall metric, **inverse purity**, which rewards grouping items together. The two metrics can be combined in an F-score.

B^3 Precision (Bagga and Baldwin, 1998): Let $L(e)$ and $C(e)$ denote the gold-standard-category and the cluster of an item e. The correctness of the relation between e and other element e' is defined as

$$\text{Correctness}(e, e') = \begin{cases} 1 & \text{iff} L(e) = L(e') \Leftrightarrow C(e) = C(e') \\ 0 & \text{otherwise} \end{cases}$$

The B^3 Precision of an item is the proportion of items in its cluster which belong to its category, including itself. The total precision is the average of the item precisions: B^3 Prec $= \text{avg}_e[\text{avg}_{e':C(e)=C(e')}\text{Correctness}(e, e')]$

B^3 Recall: is calculated in a similar way, inverting the roles of clusters and categories. The **B^3 F-score** is obtained by combining B^3 precision and B^3 recall.

4.2 Gold standards

We have built two annotated sets, one to be used as a development set for adjusting the parameters, and a second one as a test set. The evaluation settings were chosen without knowledge of the test set. Each of the two sets contains 75 instances chosen randomly from the complete set of instances with ranked attributes (Section 3.2 described the input data). For the random sampling, the instances were weighted with their frequency in the query logs as full queries, so that more frequent instances have higher chance to be chosen. This ensures that uncommon instances are not overrepresented in the gold-standard.

The annotators contributed 50 additional instances (25 for development and 25 for testing) that they considered interesting to study, e.g., because of having several salient meanings.

Five human annotators were shown the top 32 attributes for each instance, and they were asked to cluster them. We decided to start with a simplified version of the problem by considering it a hard clustering task.

Purity	Inv. Purity	F-score	B^3 Precision	B^3 Recall	B^3 F-score
0.94	0.95	0.92	0.90	0.92	0.91

Table 2: Inter-judge agreement scores.

Polysemous	Main meanings
airplane	machine, movie
apple	fruit, company
armstrong	unit, company, person
chain reaction	company, film, band, chemistry
chf	airport, currency, heart attack
darwin	person, city
david copperfield	book, performer, movie
delta	letter, airways

Table 3: Examples of polysemous instances.

Table 2 shows that the average agreement scores between judge pairs, measured with the same evaluation metrics used for the system output, are quite high. In the first three metrics, the F-score is not an average of precision and recall, but a weighted average calculated separately for each cluster, so it may have a value that is not between the values of precision and recall.

The annotated instances were classified as monosemous/polysemous, depending on wether or not they had more than one cluster with enough (five) attributes. This classification allows to report separate results for the whole set (where instances with just one major sense dominate) and for the subset of polysemous instances. Table 3 shows examples of polysemous instances. Exam-

		All instances						polysemous instances					
	Weights	**Purity**	**Inv. Purity**	**F score**	**B^3 Prec.**	**B^3 Recall**	**B^3 F score**	**Purity**	**Inv. Purity**	**F score**	**B^3 Prec.**	**B^3 Recall**	**B^3 F score**
	All-in-one	0.797	1.000	0.766	0.700	1.000	0.797	0.558	1.000	0.540	0.410	1.000	0.573
	All-singletons	1.000	0.145	0.187	1.000	0.145	0.242	1.000	0.205	0.266	1.000	0.205	0.333
	Random	0.888	0.322	0.451	0.851	0.246	0.373	0.685	0.362	0.447	0.595	0.276	0.373
Random Init.	Only snippets	0.809	0.374	0.417	0.737	0.311	0.410	0.596	0.430	0.401	0.483	0.361	0.399
	Only sessions	0.797	0.948	0.728	0.700	0.944	0.753	0.558	1.000	0.540	0.410	1.000	0.573
	Only class labels	0.798	0.983	0.760	0.701	0.969	0.785	0.561	0.990	0.541	0.415	0.981	0.574
	No snippets	0.798	0.934	0.723	0.702	0.918	0.744	0.561	0.990	0.541	0.415	0.981	0.574
	No sessions	0.809	0.374	0.417	0.737	0.311	0.410	0.596	0.430	0.401	0.483	0.361	0.399
	No class labels	0.809	0.374	0.417	0.737	0.311	0.410	0.596	0.430	0.401	0.483	0.361	0.399
	All	0.809	0.380	0.420	0.736	0.316	0.414	0.596	0.430	0.400	0.483	0.361	0.399
K-Means Init.	Only snippets	0.844	0.765	0.700	0.771	0.654	0.675	0.671	0.806	0.587	0.556	0.719	0.611
	Only sessions	0.798	0.957	0.736	0.702	0.949	0.759	0.558	1.000	0.540	0.410	1.000	0.573
	Only class labels	0.824	0.656	0.622	0.747	0.568	0.604	0.641	0.768	0.565	0.519	0.699	0.575
	No snippets	0.824	0.655	0.622	0.748	0.562	0.598	0.640	0.768	0.565	0.518	0.698	0.574
	No sessions	0.843	0.770	0.701	0.769	0.661	0.677	0.671	0.806	0.587	0.556	0.719	0.611
	No class labels	0.844	0.762	0.698	0.771	0.651	0.673	0.671	0.806	0.587	0.556	0.719	0.611
	All	0.843	0.767	0.699	0.770	0.657	0.675	0.671	0.806	0.587	0.556	0.719	0.611

Table 4: Scores over all instances and over polysemous instances.

ples of monosemous instances are *activision*, *amctheaters*, *american airlines*, *ask.com*, *bebo*, *disney* or *einstein*. 22% of the instances in the development set and 13% of the instances in the test set are polysemous.

4.3 Parameter tuning

We tuned the different parameters of the algorithm using the development set. We performed several EM runs including all three data sources, modifying the following parameters: the smoothing ϵ added to the cluster soft-assignment in the Maximization step (Manning et al., 2008), the number K of clusters for K-Means and EM, and the number k of top ranked class labels that two clusters need to have in common in order to be merged at the post-processing step. The best results were obtained with $\epsilon = 0.4$, $K = 5$ and $k = 1$. These are the values used in the experiments mentioned from now on.

4.4 EM initialization and data sources

Table 4 shows the results after running EM over the development set, using every possible combination of data sources, and the two initialization strategies (random and K-Means). Several observations can be drawn from this table:

First, as mentioned in Section 2, the evaluation metrics are biased towards the all-in-one solution. This is worsened by the fact that the majority of the instances in our dataset are monosemous. Therefore, the highest F-scores and B^3 F-scores are obtained by the all-in-one baseline, although it is not the most useful clustering.

When using only class labels, EM tends to produce results similar to the all-in-one baseline This can be explained by the limited class vocabulary which makes most of the attributes share class labels. The bad results when using only sessions are caused by the presence of attributes with no session terms, due to insufficient data.

The random clustering baseline (third line in Table 4) tends to give smaller clusters than EM, because it distributes instances uniformly across the clusters. This leads to better precision scores, and much worse recall and F-score metrics.

From these results, we conclude that snippet terms are the most useful resource for clustering. The other data sources do not provide a significant improvement over it. The best results overall for the polysemous instances, and the highest results for the whole dataset (excluding the outliers that are too similar to the all-in-one baseline) are obtained using snippet terms. For these configurations, as we expected, the K-Means initialization does a better job in avoiding local optima during EM than the random one.

4.5 Post-processing

Table 5 includes the results on the development set after post-processing, using the best configuration for EM (K-Means initialization and snippet terms for EM). Post-processing slightly hurts the B^3 F-score for polysemous terms, but it improves results for the whole dataset, as it merges many clusters for the monosemous instances.

Data	Method	Purity	Inv. Purity	F-score	B^3 Prec.	B^3 Recall	B^3 F-score
All instances	All-in-one	0.797	1.000	0.766	0.700	1.000	0.797
	All-singletons	1.000	0.145	0.187	1.000	0.145	0.242
	K-Means + EM (snippets)	0.844	0.765	0.700	0.771	0.654	0.675
	K-Means + EM (snippets) + postprocessing	0.825	0.837	0.728	0.743	0.761	0.722
Polysemous	All-in-one	0.558	1.000	0.540	0.410	1.000	0.573
	All-singletons	1.000	0.205	0.266	1.000	0.205	0.333
	K-Means + EM (snippets)	0.671	0.806	0.587	0.556	0.719	0.611
	K-Means + EM (snippets) + postprocessing	0.644	0.846	0.592	0.518	0.777	0.607

Table 5: Scores only over all and polysemous instances, without and with postprocessing.

K-Means output	EM output	Post-processing
pictures, family, logo, biography inauguration, song, lyrics, foods, quotes, timeline, shoes, health care maps, art, kids, history, speeches official website, facts, scandal economy, blog, music, flag, camping	pictures, biography, inauguration song, lyrics, foods, timeline, camping, shoes, maps, art, history, official website, facts, speeches scandal, blog, music	pictures, biography, inauguration song, lyrics, goods, timeline, camping, shoes, maps, art, history official website, facts, speeches scandal, blog, music, family, kids daughters
approval rating	approval rating, health care, economy	approval rating, health care, economy
daughters	family, kids, daughters	logo, quotes, symbol, definition religion, slogan, books, flag
symbol	logo, quotes, symbol, flag	
definition, religion, slogan, books	definition, religion, slogan, books	

Table 6: Attributes extracted for the monosemous instance *obama*, using snippet terms for EM.

4.6 Clustering examples

Tables 6 and 7 show examples of clustering results for three instances chosen as representatives of the monosemous and the polysemous subsets. These show that the output of the K-Means initialization can uncover some meaningful clusters, but tends to generate a dominant cluster and a few small or singleton clusters. EM distributes the attributes more evenly across clusters, combining attributes that are closely related.

For monosemous instances like *obama*, EM generates small clusters of highly related attributes (e.g, *family*, *kids* and *daughters*). Post-processing merges some of the clusters together, but it fails to merge all into a single cluster.

For *darwin*, two of the small clusters given by K-Means are actually good, as *ports* is the only attribute of the operating system, and *lyrics* is one of the two attributes referring to a song titled Darwin. EM again redistributes the attributes, creating two large and mostly correct clusters.

For *david copperfield*, EM creates two clusters for the performer, one for the book, one for the movie, and one for *tattoo* (off-topic for this instance). The two clusters referring to the performer are merged in the post-processing, with some errors remaining, e.g, *trailer* and *second wife* are in the wrong cluster.

4.7 Results on the test set

Table 8 show the results of the EM clustering and the postprocessing step when executed on the test set. The settings are those that produced the best results on the development set: using EM initialized with K-Means, and using only snippet terms for the generative model.

As mentioned above, the test set has a higher proportion of monosemous queries than the development set, so the all-in-one baseline produces better results than before. Still, we can see the same trend happening: for the whole dataset the F-score metrics are somewhat worse than the best baseline, given that the evaluation metrics all overvalue the all-in-one baseline, but this can be considered an artifact of the metrics. As with the development set, using EM produces the best precision scores (except for the all-singletons baseline), and the postprocessing improves precision and F-score over the all-in-one baseline. The whole system improves considerably the F-score for the polysemous terms.

5 Conclusions

This paper investigates the new task of inducing instance senses using ranked lists of attributes as input. It describes a clustering procedure based on the EM model, capable of integrating differ-

Instance	K-Means output	EM output	Post-processing
Darwin	maps, shoes, logo, awards, weather pictures, quotes, definition, jobs, tourism, biography, hotels, attractions, beaches, accommodation, tv show, clothing, postcode, music facts, review, history side effects, airlines, prices, lighting	maps, shoes, logo, weather jobs, tourism hotels, attractions, beaches, accommodation, tv show, clothing, postcode, music, review side effects, airlines, prices, lighting	maps, shoes, logo, weather jobs, tourism hotels, attractions, beaches, accommodation, tv show, clothing, postcode, music, review side effects, airlines, prices, lighting definition, population
	ports	awards, ports	awards, ports
	evolution, theory, books	evolution, theory, quotes pictures, biography, facts, history, books	evolution, theory, quotes pictures, biography, facts, history, books
	lyrics	lyrics	lyrics
	population	definition, population	
David Copperfield	summary, biography, pictures, quotes, strokes, book review, tricks, tour dates, characters, lyrics, plot, synopsis, dating, logo, themes, author, filmography, cast members, official website, trailer, setting, religion	biography, pictures, quotes, strokes, tricks, tour dates, lyrics, dating, logo, filmography, cast members, official website, trailer, setting, religion	biography, pictures, girlfriend quotes, strokes, tricks, tattoo tour dates, secrets, lyrics, wives, music, dating, logo, filmography, blog, cast members, official website, trailer, setting, religion
	house, reviews	book review, review, house, reviews	book review, review, house, reviews
	tattoo	tattoo	summary, second wife, characters, plot, synopsis, themes, author
	second wife	summary, second wife, characters, plot, synopsis, themes, author	
	girlfriend, secrets, wives, review, music, blog	girlfriend, secrets, wives, music, blog	

Table 7: Attributes extracted for three polysemous instances, using snippet terms for EM.

Set	Solution	Purity	Inverse Purity	F-score	B^3 Precision	B^3 Recall	B^3 F-score
All	All-in-one	0.907	1.000	0.892	0.858	1.000	0.908
	All-singletons	1.000	0.076	0.114	1.000	0.076	0.136
	Random	0.936	0.325	0.463	0.914	0.243	0.377
	EM	0.927	0.577	0.664	0.896	0.426	0.561
	EM+postprocessing	0.919	0.806	0.804	0.878	0.717	0.764
Polysemous	All-in-one	0.588	1.000	0.586	0.457	1.000	0.613
	All-singletons	1.000	0.141	0.210	1.000	0.141	0.239
	Random	0.643	0.382	0.441	0.549	0.288	0.369
	EM	0.706	0.631	0.556	0.626	0.515	0.547
	EM+postprocessing	0.675	0.894	0.650	0.564	0.842	0.661

Table 8: Scores in the test set.

ent data sources, and explores cluster initialization and post-processing strategies. The evaluation shows that the most important of the considered data sources is the snippet terms obtained from search engine results to queries made by concatenating the instance and the attribute. A simple post-processing that merges attribute clusters that have common class labels can improve recall for monosemous queries. The results show improvements across most metrics with respect to a random baseline, and F-score improvements for polysemous instances.

Future work includes extending the generative model to be applied across the board, linking the clustering models of different instances with each other. We also intend to explore applications of the clustered attributes in order to perform extrinsic evaluations on these data.

References

Agirre, Eneko and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of SemEval-2007*, pages 7–12. Association for Computational Linguistics.

Amigó, E., J. Gonzalo, J. Artiles, and F. Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Artiles, Javier, Julio Gonzalo, and Satoshi Sekine. 2007. The semeval-2007 WePS evaluation: Establishing a benchmark for the web people search task. In *Proceedings of SemEval-2007*, pages 64–69.

Artiles, J., J. Gonzalo, and S. Sekine. 2009. Weps 2 evaluation campaign: overview of the web people search clustering task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

Bagga, A. and B. Baldwin. 1998. Entity-based cross-document co-referencing using the vector space model, Proceedings of the 17th international conference on Computational linguistics. In *Proceedings of ACL-98*.

Banko, M., Michael J Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the Web. In *Proceedings of IJCAI-07*, pages 2670–2676, Hyderabad, India.

Bellare, K., P.P. Talukdar, G. Kumaran, F. Pereira, M. Liberman, A. McCallum, and M. Dredze. 2007. Lightly-Supervised Attribute Extraction. In *NIPS 2007 Workshop on Machine Learning for Web Search*.

Brody, Samuel and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of EACL '09*, pages 103–111.

Cafarella, M.J., A. Halevy, D.Z. Wang, and Y. Zhang. 2008. Webtables: Exploring the Power of Tables on the Eeb. *Proceedings of the VLDB Endowment archive*, 1(1):538–549.

Cui, G., Q. Lu, W. Li, and Y. Chen. 2009. Automatic Acquisition of Attributes for Ontology Construction. In *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages*, pages 248–259. Springer.

Dempster, A.P., N.M. Laird, D.B. Rubin, et al. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Etzioni, O., M. Banko, S. Soderland, and S. Weld. 2008. Open Information Extraction from the Web. *Communications of the ACM*, 51(12), December.

Gao, W., C. Niu, J. Nie, M. Zhou, J. Hu, K. Wong, and H. Hon. 2007. Cross-lingual query suggestion using query logs of different languages. In *Proceedings of SIGIR-07*, pages 463–470, Amsterdam, The Netherlands.

Geiss, J. 2009. Creating a Gold Standard for Sentence Clustering in Multi-Document Summarization. *ACL-IJCNLP 2009*.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING-92*, pages 539–545, Nantes, France.

Mann, Gideon S. and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of HLT-NAACL 2003*, pages 33–40. Association for Computational Linguistics.

Manning, C.D., P. Raghavan, and H. Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press New York, NY, USA.

Nastase, V. and M. Strube. 2008. Decoding wikipedia categories for knowledge acquisition. In *Proceedings of AAAI-08*, pages 1219–1224, Chicago, Illinois.

Paşca, M. and B. Van Durme. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of IJCAI-07*, pages 2832–2837, Hyderabad, India.

Paşca, M. and B. Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of ACL-08*, pages 19–27, Columbus, Ohio.

Paşca, M., B. Van Durme, and N. Garera. 2007. The role of documents vs. queries in extracting class attributes from text. In *Proceedings of CIKM-07*, pages 485–494, Lisbon, Portugal.

Paşca, M. 2007. Organizing and searching the World Wide Web of facts - step two: Harnessing the wisdom of the crowds. In *Proceedings of WWW-07*, pages 101–110, Banff, Canada.

Ravi, S. and M. Paşca. 2008. Using Structured Text for Large-Scale Attribute Extraction. In *CIKM*. ACM New York, NY, USA.

Sekine, S. 2006. On-Demand Information Extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics Morristown, NJ, USA.

Silverstein, C., H. Marais, M. Henzinger, and M. Moricz. 1999. Analysis of a very large web search engine query log. In *ACM SIGIR Forum*, pages 6–12. ACM New York, NY, USA.

Suchanek, F., G. Kasneci, and G. Weikum. 2007. Yago: a core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of WWW-07*, pages 697–706, Banff, Canada.

Tokunaga, K., J. Kazama, and K. Torisawa. 2005. Automatic discovery of attribute words from Web documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 106–118, Jeju Island, Korea.

Wong, T.L. and W. Lam. 2009. An Unsupervised Method for Joint Information Extraction and Feature Mining Across Different Web Sites. *Data & Knowledge Engineering*, 68(1):107–125.

Wu, F. and D. Weld. 2008. Automatically refining the Wikipedia infobox ontology. In *Proceedings of WWW-08*, pages 635–644, Beijing, China.

Wu, F., R. Hoffmann, and D. Weld. 2008. Information extraction from Wikipedia: Moving down the long tail. In *Proceedings of KDD-08*, pages 731–739, Las Vegas, Nevada.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL-95*, pages 189–196. Association for Computational Linguistics.

Yoshinaga, N. and K. Torisawa. 2007. Open-Domain Attribute-Value Acquisition from Semi-Structured Texts. In *Proceedings of the Workshop on Ontolex*, pages 55–66.

Zhao, Y. and G. Karypis. 2002. Criterion functions for document clustering. Technical report, Experiments and Analysis University of Minnesota, Department of Computer Science/Army HPC Research Center.