# Bootstrapping information extraction mappings by similarity-based reuse of taxonomies

Martijn Spitters, Remko Bonnema, Mihai Rotaru, and Jakub Zavrel

Textkernel BV
Nieuwendammerkade 28A-17
1022AB, Amsterdam, The Netherlands
{spitters,bonnema,rotaru,zavrel}@textkernel.nl

**Abstract.** Many practical information extraction systems use simple taxonomies for mapping extracted strings to client-specific concept codes. In such taxonomies, concepts are defined as groups of semantically similar words and phrases. For the mapping to be accurate, a new client-specific taxonomy – usually nothing more than a set of concept codes, each with a single description – needs to be enriched with the domain-specific terminology variations, which is a very labor-intensive task. In this paper, we describe a method to significantly reduce the required manual effort for this task. Our approach is based on combining multiple existing client-specific taxonomies into a single semantic space. On a set of gold standard taxonomies our method achieves an average precision of 91% and a recall of 55%. An additional practice test shows that the method saves at least 62% of the manual effort needed to enrich a new taxonomy.

## 1 Introduction & motivation

For the effective deployment of generic information extraction (IE) systems, a module for mapping extracted strings to client-specific codes (e.g. map 'cardiologist' and 'cardiovascular specialist' to code '59C' – see Figure 1) is indispensable. Typically, simple (non-hierarchical) taxonomies are used to guide this process, which we will refer to as **normalization**. The type of taxonomy that is used in our IE systems starts with an unstructured set of **concepts**. In its most basic form, a concept consists of a **concept code** (simply some unique client-specific identifier of the concept) and a single textual description called the **concept description**. Typical examples of concept descriptions that can be found in for instance a taxonomy of job titles, are 'account manager', 'software engineer', and 'administrative assistant'.

Normalization is usually based on string similarity: an extracted string is mapped to the concept of the closest matching concept description in the taxonomy. However, because a certain concept can usually be expressed in (many) different ways, it cannot be captured in a single description, but rather reflects a *group* of *semantically similar* descriptions, or **variants**. Partly, such a group of variants consists of synonyms (e.g. 'shop assistant' and 'retail assistant') and abbreviations (e.g. 'ceo' and 'chief executive officer'), but depending on the taxonomy's characteristics it can also contain near-synonyms or hyponyms (e.g. 'software engineer', 'programmer', and 'sr. Java/J2EE developer').

The client-specific taxonomies represent partially similar and partially different conceptualizations of the domain, each with its own specific *coverage* and *granularity*. As each taxonomy is a certain clustering of domain-specific terminology, it can be regarded as a particular semantic partitioning, or *view* of the domain. Two strings that are regarded as description variants of the same concept in one taxonomy, might well be assigned to different concepts in another, more fine-grained taxonomy. As an illustration, Figure 1 shows different conceptualizations of healthcare-related job titles, taken from three real-world job title taxonomies.

| Taxonomy 1 | Taxonomy 2 | Taxonomy 3 |
|---|---|---|
| **59C Cardiologist** | **4138 Specialist (Medical)** | **CVG-12 Health/Care** |
| analyst cardiology | surgeon | pharmasist |
| heart specialist | medical specialist | pharmacy assistent |
| medical specialist cardiology | cardiologist | druggist |
| cardiothoracic surgeon | gynaecologist | undertaker |
| cardiovascular specialist | oncologist | general practitioner |
| cardiac surgeon | dermatologist | nurse |
| **60I Internist** | internist | nursing assistant |
| internal medicine | **4139 Nursing/Care** | geriatric nurse |
| dr of internal medicine | nurse | medical specialist |
| **61D Dermatologist** | healthcare professional | internist |
| pediatric dermatologist | mental health nurse | clinical analyst |
| skin disease specialist | psychiatric nurse | surgeon |
| specialist dermatology | cert. nursing assistant | gynaecologist |
| **62P Pediatrician** | nurse anesthetist | dermatologist |
| paediatric specialist | clinical nurse specialist | trauma surgeon |
| youth health specialist | elderly care nurse | physician |

**Fig. 1.** Fragments of three different conceptualizations of health/medical-related job titles. The concept codes and the concept descriptions are bold.

For accurate normalization of extracted phrases, a new taxonomy has to be expanded with the typical terminology variations of its concepts – a process we call **enrichment**. Because each client has a unique semantic view on the domain, enriching a taxonomy is hard to automate. How exactly a particular taxonomy should be enriched depends on its coverage and granularity, and therefore typically requires human assessment. Obviously, manually enriching hundreds of concepts with thousands of description variants, is a very labor-intensive task.

In this paper, we present a method to significantly reduce the manual effort for enriching new taxonomies with variants of their concept descriptions. The method exploits multiple manually created taxonomies by representing their partially shared description variants in a single semantic space. Our approach comprises two steps: (1) a string similarity-based step for selecting attractor descriptions for the concepts in the new

taxonomy, and (2) a step for linking the remaining description variants to the attractors based on their semantic similarity. This second step seeks to exploit the semantic knowledge present in the *combination* of the existing taxonomies.

The remainder of this paper is organized as follows. First, we will describe the two steps of our enrichment method. In section 4 we will briefly look at the data used in our experiments, as well as our evaluation method. Section 5 describes the experiments we conducted along with a discussion of their results. Section 3 places our work in the context of existing research, and finally, in section 6 we list our conclusions.

## 2 Methodology

The starting point of our approach is a set of manually enriched taxonomies for the same domain. These taxonomies – each one different in coverage and granularity – are represented as a sparse matrix of which the rows represent the description variants in the taxonomies and a column represents their concept assignments in a particular taxonomy. In other words, a row of the matrix is a sparse vector of concept codes. We will refer to a representation of a description variant in the matrix as an **instance**. Assuming that the more different semantic contexts two instances share, the more semantically related they are, row similarities in the matrix can be regarded as semantic similarities.

Given a new taxonomy – i.e. a never before seen conceptualization of the domain – the goal is to *automatically enrich its concepts with as much description variants from the matrix as possible*. This task can be regarded as a weakly-supervised classification task. For each instance in the matrix, we want to predict a single (or no) class label from a new, unknown class set (the concepts of the new taxonomy), based on existing assignments from other class sets (the manually enriched taxonomies, represented in the matrix). However, in order to do this, we need some initial mapping from our instance space to the new class set which can function as a seeding set. Section 2.1 describes the first step of the enrichment process, in which instances, called **attractors**, are linked to the concepts of the new taxonomy based on string similarity, and which results in a seeding set (i.e. a set of instances labeled with the target concepts). The second step of our approach attempts to link the remaining instances in the matrix to these attractors, based on the similarity of their semantic vectors. Section 2.2 goes into detail about this step.

### 2.1 Step 1: Attractor selection

The first step of our method attempts to select representatives in the matrix for the concepts of the new taxonomy based on string similarity, which we will call **attractors** (as they will be used to 'attract' semantically similar instances in the second step). Each instance in the matrix is compared to each concept of the new taxonomy. An instance is linked to the best matching concept if their string similarity exceeds a certain threshold. The left hand side of Figure 2 shows the attractor selection step.

For the sake of clarity, we will also illustrate the individual steps of our method using the taxonomy fragments in Figure 1. Let us assume that the description variants

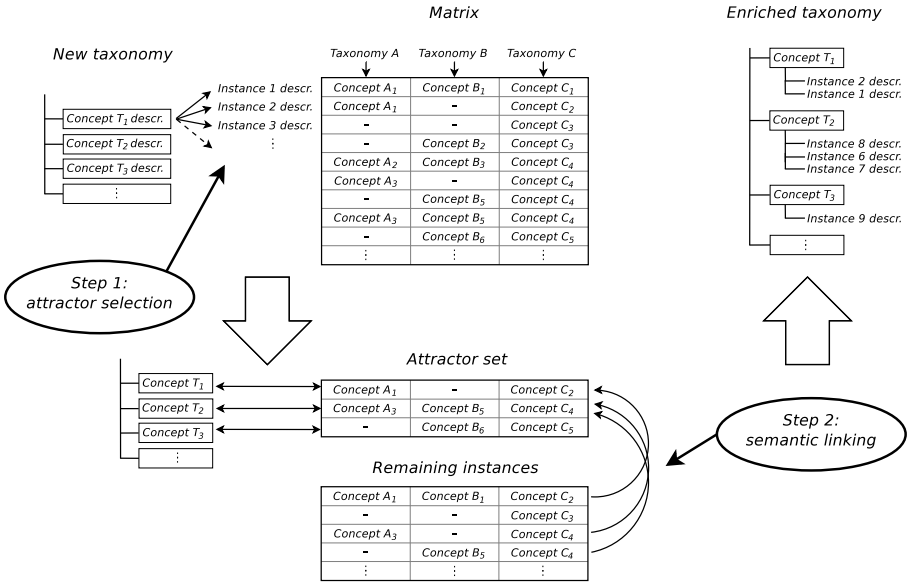


**Fig. 2.** Visualization of the two-step taxonomy enrichment process.

in the three fragments are used to form the matrix. For example, the description variant *dermatologist* will be represented as matrix instance '**61D, 4138, CVG-12**'. Given a new taxonomy with two medical concepts: '**023CRD Cardiology**' and '**024OMS Other medical specialisms**', our goal is to find for each of these concepts the most suitable attractor available in the matrix. Because we only use string similarity in this step, we will find *cardiologist* , represented in the matrix as instance '**59C, 4138, -**' for the first concept, and *medical specialist* (instance '**-, 4138, CVG-12**') for the second concept.

Step 1 results in a split of the matrix into a set of linked attractors (the **attractor set**) and a set of instances that could not yet be linked to any concept. From a classification point of view, the attractor set is the seeding set for step 2. Note that it is possible that for some of the concepts of the new taxonomy no attractor will be found, as an instance and a concept are only linked if their string similarity exceeds a fixed threshold. Also, multiple attractors can be linked to a single concept. The attractor coverage of a new taxonomy after step 1 relies heavily on the richness of the matrix (the description variety), the string similarity threshold, and also the quality of the concept descriptions of the new taxonomy. Because the second step of our method depends solely on the outcome of step 1, it is essential to mimimize the number of errors in the first step. An incorrectly linked attractor potentially attracts (many) new incorrect instances.

For optimizing the attractor selection accuracy, we experimented with several different string similarity metrics, as well as linguistic preprocessing (lemmatization and morpheme splitting) of the descriptions. The details of these experiments and their results will be described in section 5.1.

### 2.2 Step 2: Semantic linking

The linking of attractors to concepts in the new taxonomy is solely based on the characteristics of their descriptions. Because semantically similar information can be expressed in many different ways (synonyms, abbreviations, hyponyms, etc. – see section 1), using only string similarity will not yield a useful taxonomy enrichment. However, provided that the matrix used for enrichment is rich and contains close or even exact matches for most concept descriptions in the new taxonomy, step 1 will give us a good basis to build upon.

As stated earlier in this paper, a row vector of the matrix encodes the combined semantic contexts of a certain description variant. A high similarity between two of these vectors means that the description variants they represent have a strong semantical relationship. As the right hand side of Figure 2 illustrates, we will use the semantic similarities between the remaining instances in the matrix and the attractor instances that were linked in step 1 to further enrich the target taxonomy. Analogous to step 1, in step 2 we compute the similarity of each remaining instance to all attractors. An instance is linked to the best matching attractor if their semantic similarity exceeds a certain threshold.

Let us return to the example we used to illustrate the first step of our method in section 2.1. In our example, we linked the attractor instance '**59C, 4138, -**' (*cardiologist*) to concept '**023CRD Cardiology**', and instance '**-, 4138, CVG-12**' (*medical specialist*) to concept '**024OMS Other medical specialisms**' of the new taxonomy. In the second step we want to link the remaining instances (i.e. *heart specialist* ('**59C, -, -**'), *dermatologist* ('**61D, 4138, CVG-12**'), etc.) to the most appropriate concepts of the new taxonomy. In order to do that, we need to compare them with the attractors that were linked to the concepts in step 1. For example, the instance of *dermatologist* is equal to that of *medical specialist*, which is the attractor of '**024OMS Other medical specialisms**'. Therefore, *dermatologist* will be linked to that concept with high confidence. Another example, *heart specialist*, will be linked to concept '**023CRD Cardiology**' because its instance is more similar to the attractor *cardiologist* that to the attractor *medical specialist*. However, whether this instance is actually linked to the concept, depends on the semantic similarity threshold that has been set in the system.

We investigated the effect of using different vector similarity metrics for this step and experimented with weighted overlap, based on the granularity of the matrix taxonomies. Section 5.2 will go into detail about these experiments.

## 3 Related work

Our research problem is partially related to the *ontology matching/mapping* problem: discovering mappings between different ontologies/taxonomies of the same domain. Mainly because of the emergence of numerous different meta-data schemas on the Internet (e.g. musical genres, photo tags), this problem is regarded as one of the most urgent problems facing the Semantic Web.

Our work differs from existing research in several aspects. Firstly, at the problem level, we are interested in mapping variants to concepts rather than concepts to concepts. Secondly, since our taxonomies have a flat structure (i.e. no hierarchy), we can

not use the structural information that is central to previous work on ontology alignment (e.g. Resnik [9] measures semantic similarity in IS-A taxonomies, based on the information two concepts share in common). Thirdly, we exploit domain information implicitly through the use of multiple manually enriched taxonomies. In contrast, previous work relies on statistics derived from external corpora [7, 5], or search results from the Internet [3].

The techniques used in our work are also explored in other ontology matching work: string similarity [10], various metrics for semantic similarity between concepts [4] and combinations thereof [11]. In terms of exploiting multiple taxonomies, in [1] ontologies of poor semantics are matched by using an intermediate background ontology.

## 4 Data & evaluation

For our experiments we used a set of Dutch work experience taxonomies. In total, we used 21 manually enriched taxonomies which are actually used for normalization in a live resume extraction system. Each taxonomy is different in nature: some of them only cover a specific domain of job titles (like Finance or Commerce), while others cover the full job spectrum. Besides their coverage, the taxonomies also differ in granularity: in some taxonomies, the description instances are distributed over a rather coarse concept division (e.g. 'Administration', 'Finance', 'Sales', 'ICT', etc.), while other taxonomies are very dense. Some taxonomies are dense for a certain domain of interest, and use a couple of general concepts to cover the rest. Also, many of the description instances do not occur in each taxonomy. The number of concepts in the taxonomy ranges from 33 to 4309, and the number of description variants from 343 to 9662.

Only 12 of the 21 taxonomies could be used for development and testing purposes, because the other 9 contained too many unusable concept descriptions (e.g. descriptions that combine several domains, like 'Advertising/Communication/Marketing/PR', or that contain additional meta-information, like 'Typist/Secretary (not executive, medical; lower education)'). From the 12 usable taxonomies, 3 were used for development purposes, and 9 were used for testing.

We used leave-one-out cross-validation to evaluate our taxonomy enrichment method according to the following procedure:

For each test taxonomy $T$:

given:
- $T_{empty}$ : non-enriched version of $T$
- $T_{gold}$ : gold standard of $T$
- $T_{enriched}$ : enriched version of $T$

do:
1. Create matrix $M_{without-T}$ of all other taxonomies
2. Enrich $T_{empty}$ with instances from $M_{without-T}$
3. Compare $T_{enriched}$ with $T_{gold}$

Several taxonomies contain a special concept (e.g. 'Other'/'Unknown') that groups jobs that were outside the scope of that particular client. We discarded these concepts in the evaluation because they introduce an non-informative semantic similarity.

Creating a matrix of 22.600 unique instances took 4 seconds on a modern pc (2 2.8GHz processors; 4G of RAM). Enriching a new taxonomy with 500 concepts with the instances from that matrix takes approximately 2 minutes.

## 5 Experiments & results

This section describes all experiments we conducted and their results. The experiments we performed to optimize the string-based attractor selection step are reported in 5.1. Subsequently, in 5.2 we will describe our semantic linking experiments, and finally, in 5.3, we will report the results of normalizing a batch of job titles extracted from resumes, using the automatically enriched taxonomies versus their manually enriched counterparts.

### 5.1 Step 1

For our attractor selection step, we experimentally compared a selection of both character-based and token-based string distance (or similarity) metrics.

As a baseline, we applied the well-known *Levenshtein distance*, an edit distance function which assigns a unit cost to all edit operations (insertion, deletion, and substitution). We also considered a method which is very similar to Levenshtein, originating from the bioinformatics community, called *Needleman-Wunsch* [8]. Needleman-Wunsch is basically Levenshtein with an additional variable cost adjustment for gaps, i.e. an insert or deletion.

The descriptions in our taxonomies are relatively short, typically consisting of one to four words. We looked at metrics that are often used by the record linkage community for matching relatively short strings like names or addresses [2]. In the record linkage literature, good results have been obtained using variants of the *Jaro* metric [6], which involves both the number and order of common characters in the similarity computation of two strings. We also considered a variant of Jaro by Winkler [12], which integrates the length of the longest common prefix of two strings.

Another method for approximate matching is *q-gram* overlap. This method chops a string in fixed-length parts of $q$ characters. The similarity between two strings is defined as the number of common q-grams over the union of all q-grams in both strings. In our experiments we used $q$=3.

A concept description can also be regarded as a bag of words. We evaluated several token-based distance metrics. The *Dice coefficient* between two word sets $X$ and $Y$ is $\frac{2|X \cap Y|}{|X+Y|}$. A similar metric in which a small number of common tokens is penalized slightly more than in the Dice coefficient is the *Jaccard coefficient*, which is simply $\frac{|X \cap Y|}{|X \cup Y|}$. A third token-based measure we applied is the *cosine similarity*, which is widely used in the information retrieval community. For most of the previously mentioned string metrics we used the SIMMETRICS open source Java library of similarity metrics, developed by Sam Chapman.

Figure 3 compares all similarity metrics described above in terms of precision and recall. We computed the micro-averaged[1] precision and recall, as it gives equal weight to each instance rather than to each concept (macro-averaged). We think this is a more appropriate measure for our enrichment system, because in a practical normalization module, each description variant in the taxonomy has equal weight as well.

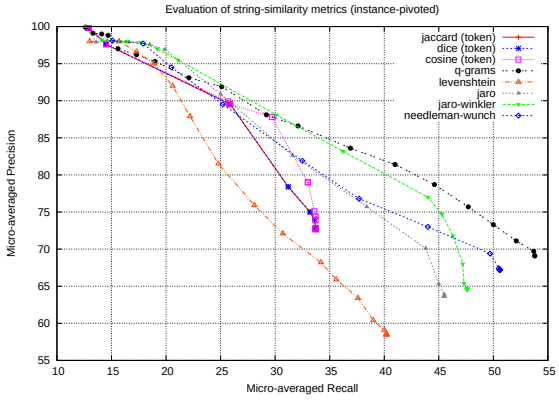


**Fig. 3.** Comparison of different string similarity metrics for the attractor selection step.

As Figure 3 shows, the Levenshtein distance is by far the worst performing metric, already dropping below 85% precision at a recall of 23% and below 60% at a recall of 40%. In the high-precision region (92-98%) the Jaro and Jaro-Winkler metrics perform best, while $q$-grams have the highest precision at recall values higher than 35%. At 100% precision, all metrics perform similarly with a recall of 12-13%.

In an attempt to improve the recall of the attractor selection step, we applied two forms of linguistic preprocessing to our descriptions: lemmatization and morpheme splitting. Especially for Dutch, which contains many compounds, these types of preprocessing can be very useful. Unfortunately, the preprocessing steps resulted only in marginal recall improvements, and just for the token-based string metrics.

**Threshold optimization** The quality of our final enrichment result partly depends on optimal threshold settings in the first step. After all, in step 2, each incorrect attractor potentially attracts (many) new incorrect descriptions to the taxonomy concept it is linked to. On the other hand, we obviously need an acceptable recall, i.e. the percentage of taxonomy concepts for which an attractor can be found.

Simply selecting the threshold at which the highest harmonic mean was measured is not optimal for our situation, because for most string similarity metrics, the highest F-scores were measured at precision values of 60-70%, which is simply too low for

[1] Micro-averaged values are calculated by constructing a global contingency table and then calculating precision and recall using these sums, whereas in macro-averaging precision and recall are calculated for each category and then the average of these is taken.

our purposes. Instead, we optimized the thresholds for the similarity metrics based on the macro-averaged precision (on the development set) of the result of step 1, where a concept for which no attractor was selected was counted as 0. This way, we optimize on a measure which takes the maximum coverage into account, and which in our view can be used as a predictor for the best possible overall enrichment. We empirically validated this hypothesis on our development set and found that indeed it generally holds. Table 1 lists this optimal macro precision value and the associated threshold for all string similarity metrics (including some combinations).

**Table 1.** Threshold optimization for string similarity metrics. The second column shows the maximum coverage precision, the third column the selected threshold.

| **Metric** | $Max.Cov.Precision$ | Thr | $Precision$ | $Recall$ | $F_1$ |
|---|---|---|---|---|---|
| Jaro-Winkler | 81.0 | 0.95 | 95.5 | 21.1 | 33.4 |
| Jaro | 81.0 | 0.90 | 96.9 | 19.9 | 31.8 |
| Needlem.-Wunsch | 81.0 | 0.85 | 94.5 | 20.5 | 32.5 |
| Jaccard+Q-Grams | 79.9 | 0.55 | 94.3 | 19.6 | 31.2 |
| Q-Grams | 79.8 | 0.65 | 95.3 | 19.0 | 30.3 |
| Levenshtein | 79.4 | 0.80 | 95.0 | 18.9 | 30.2 |
| Cosine+Q-Grams | 78.9 | 0.60 | 94.3 | 18.8 | 30.1 |
| Cosine | 77.7 | 0.80 | 97.7 | 14.5 | 23.8 |
| Dice | 77.7 | 0.80 | 97.7 | 14.5 | 23.8 |
| Jaccard | 77.7 | 0.65 | 97.7 | 14.5 | 23.8 |

### 5.2 Step 2

In step 2, we want to compare the semantic vectors of the remaining matrix instances with those of the attractors selected in step 1. Each position of an instance vector points to a particular taxonomy and its value can be one of all concept id's of that taxonomy, or the undefined value $\omega$ if the concerning instance does not occur in that taxonomy.

The most basic distance metric that works for vectors with symbolic features is the $L_1$ distance, also referred to as *Hamming distance*, *Manhattan distance*, *city-block distance*, or the *overlap metric*. The $L_1$ distance between two vectors $x$ and $y$ is calculated as in equations 1 and 2.

$$L_1 = \sum_{i=1}^{n} \delta(x_i, y_i) \tag{1}$$

where:

$$\delta(x_i, y_i) = \begin{cases} 1 \text{ if } x_i \neq y_i, \\ 0 \text{ if } x_i = y_i. \end{cases} \tag{2}$$

To get a distance between 0 and 1, we normalize the $L_1$ scores by the vector length (i.e. the number of taxonomies in the matrix).

A distance of 0 means that two instances are equal. To handle the undefined values a bit more elegantly, we also applied a variant of 2, given in equation 3.

$$\delta(x_i, y_i) = \begin{cases} 1 \text{ if } x_i \neq y_i, \\ 0 \text{ if } x_i = y_i, \\ 0.5 \text{ if } x_i = \omega \vee y_i = \omega. \end{cases} \tag{3}$$

This variation assigns a value of 0.5 if one or both of the instances we are comparing is undefined in the concerning taxonomy. Intuitively, this seems like a more appropriate scoring method, because if an instance is not defined in a taxonomy, it means we do not have any information about it, i.e. we do not know whether or not it would be assigned to the same concept as the instance we are comparing it with.

Figure 4 shows the performance of some $L_1$ variations compared to the baseline of applying only step 1 (Jaro-Winkler). The graph shows the significant recall contribution of step 2 in the enrichment process. The variation of $L_1$ described in equation 3 has slightly better recall than the default version (equation 2) in the +91% precision region, but decreases recall at precision values of 90% and lower. A variation in which we combined the $L_1$ distance with a string distance performs worse in the +85% precision region, but better at lower precision values. Table 2 lists the best overall performing metric combinations.

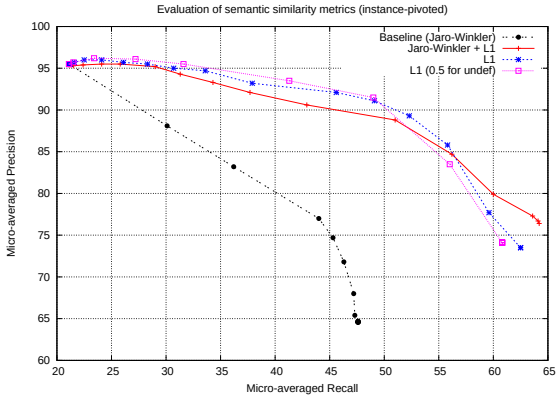


**Fig. 4.** Three $L_1$ variations compared to the baseline of only applying step 1.

**Table 2.** Best-performing metric combinations.

| *Step 1 metric* | *Step 2 metric* | $Max\ F_{0.5}$ | $Precision$ | $Recall$ | $R@P = 90\%$ | $R@P = 95\%$ |
|---|---|---|---|---|---|---|
| Jaro | $L_1$ (0.5 for undef) | 75.7 | 84.3 | 55.7 | 51.2 | 38.2 |
| | $L_1$ | 74.9 | 86.0 | 54.7 | 51.2 | 34.1 |
| Jaro-Winkler | $L_1$ (0.5 for undef) | 75.4 | 83.5 | 56.0 | 51.5 | 34.2 |
| | $L_1$ | 75.1 | 85.4 | 55.8 | 51.2 | 30.7 |

**Granularity-based weighing** The granularity of a taxonomy has a direct relation with the mutual semantic similarity of description variants that belong to the same concept. Roughly said, the more specific a taxonomy concept is, the more likely it is that the description variants of that concept are actual synonyms in the linguistic sense. In the type of taxonomies we work with, description variants of a specific concept are thus semantically more similar than the variants ranked under a broad concept.

We considered a granularity-based weighted version of the $L_1$ metric (equation 1 and 2). The $x_i \neq y_i$ penalty was multiplied by the granularity of taxonomy $T_i$. This granularity was computed by dividing the number of description instances in $T_i$ by the number of concepts in $T_i$. The weights were normalized to values between 0 and 1. By applying these weights, a concept match in a taxonomy with broad, highly variational concepts will be rewarded less than a concept match in a very specific, dense taxonomy. Figure 5 shows the result of this experiment. The weighted $L_1$ version performs better than the default version over the entire precision range. At 91% precision, recall is improved with 5.2%.

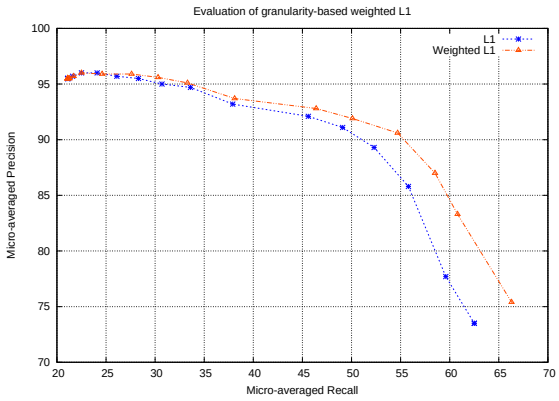


**Fig. 5.** The effect of granularity-based weighing on enrichment performance.

### 5.3 Normalization accuracy

It is not quite clear how the enrichment performance relates to normalization performance. That is, because certain description variants rarely occur in input documents and others occur very frequently, an enrichment precision of 91% does not necessarily mean that this taxonomy, when used for normalization in a practical IE system (see section 1), will map 9% of the input phrases to the wrong concept. Moreover, a recall of 55% might seem low, but because normalization is based on string-similarity, many description variants, like the plural form or orthographic variations of a description, are actually redundant.

We compared the normalization performance of taxonomies that were enriched automatically with that of their manually enriched versions. From the 500 most frequently

extracted job title descriptions in a batch of Dutch resumes, we randomly selected 200. These descriptions were normalized with three different taxonomies.

**Table 3.** Normalization accuracy of three taxonomies; manually versus automatically enriched.

| **Enrichment** | Acc. $T_1$ | Acc. $T_2$ | Acc. $T_3$ |
|---|---|---|---|
| Baseline | 0.49 | 0.61 | 0.51 |
| Manual | 0.76 | 0.78 | 0.93 |
| Auto | 0.68 | 0.73 | 0.70 |
| Relative improvement | 70% | 71% | 45% |

Table 3 shows that the accuracy of the baseline (normalization with the empty taxonomy, using only the original concept descriptions) lies around 50-60%. The manually enriched versions perform significantly better, especially for $T_3$, which is a very large, well-maintained taxonomy. The third row shows the performance with the automatically enriched taxonomies, which is around 70% for all three taxonomies. The relative improvement at the bottom of the table is the percentage of the performance gap between the baselines and the manually enriched versions that our enrichment method closes, which on average is 62%. The relative contribution of auto-enrichment is only 45% for $T_3$, which is probably due to the fine-grainedness of that particular taxonomy. Because of the specificity of the concept descriptions of the non-enriched version, it is harder to find attractors in step 1 that exceed the strict threshold. We re-enriched this taxonomy with an adapted threshold for attractor selection (step 1) as well as for semantic linking (step 2). The resulting new enrichment performs 4% better than the more stricter enrichment. This tells us that the thresholds of our method should be fine-tuned for different granularity levels.

## 6 Conclusions

In this paper we introduced a novel method for enriching taxonomies with lexical variants of their concept descriptions. The method relies on a combined representation of existing taxonomies, and is composed of the following two steps:

1. high-precision selection of attractor descriptions for concepts based on string similarity;
2. linking of description variants to attractors, using semantic similarity based on their concept assignments in existing taxonomies.

The best version of our system uses the Jaro-Winkler metric for attractor selection and a granularity-based weighted $L_1$ distance for semantic linking in step 2, and enriches a new taxonomy with a precision of 91% at a recall of 55% (measured by cross-validation with nine taxonomies of different coverage and granularity). Obviously, this will save at least 55% of our manual enrichment work. However, our additional normalization test has shown that in practice, this number is actually 62%, and for some taxonomies the savings can be as high as 70%.

**Acknowledgements** The research described in this paper was conducted within the DAESO project (http://daeso.uvt.nl) funded by the Stevin program (De Nederlandse Taalunie).

## References

1. Zharko Aleksovski, Michel Klein, Warner ten Kate, and Frank van Harmelen: Matching unstructured vocabularies using a background ontology. ISWC Ontology Matching Workshop (2006)
2. William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg: A Comparison of String Distance Metrics for Name-Matching Tasks. Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03) (2003)
3. Risto Gligorov, Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen: Using Google Distance to weight approximate ontology matches. Proceedings of the 16th international conference on World Wide Web, New York, USA (2007)
4. Ryutaro Ichise, Hiedeaki Takeda, and Shinichi Honiden: Integrating multiple internet directories by instance-based learning. Proceedings of IJCAI (2003)
5. Aminul Islam and Diana Inkpen: Semantic similarity of short texts. Proceedings of RANLP, pp. 291-297 (2007)
6. M.A. Jaro: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association 84:414-420 (1989)
7. Jay J. Jiang and David W. Conrath: Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings of ROCLING X, Taiwan (1997)
8. Saul B. Needleman and Christian D. Wunsch: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48 (3): 443-53 (1970)
9. Philip Resnik: Using information content to evaluate semantic similarity in a taxonomy. Proceedings of IJCAI-95 pp. 448-453, Montreal, Canada (1995)
10. Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias: A String Metric for Ontology Alignment. Proceedings of ISWC, pp. 624-637 (2005)
11. Michael Wick, Khashayar Rohanimanesh, Andrew McCallum, and AnHai Doan: A discriminative approach to ontology mapping. *International Workshop on New Trends in Information Integration (NTII) at VLDB WS* Auckland, New Zealand (2008)
12. W.E. Winkler: The state of record linkage and current research problems. Statistics of Income Division, Internal Reveneu Service Publication R99/04. Available from http://www.census.gov/srd/www/byname.html (1999)