

# From hyperlinks to semantic links

## A Probabilistic Approach

### ABSTRACT

In this paper, we propose an approach for detecting the implicit relations between 2 entities.

### 1. INTRODUCTION

Semantifying web documents is a huge task. Recently, more and more hyperlinked documents emerge. There are almost semi-structured information everywhere, but unfortunately the links lack an explanation of why they are connected, and why human may consider them related. The types of hyperlink are article to article, article to knowledge base (or encyclopedia-like databases). Each link has two ends: the original one (the web page/entity page contains it) and the oriented one (the web page/entity page it points to), and each end has 2 possible types, an anchor url or an anchor entity. Hence, in all there are 4 possible combinations, which consist the 4 type of links. An anchor entity can be named entities such as movies, products, people names, locations etc. In this case we explain the relation between the entity and the original article. As for an anchor url, an automatic target entity extraction process [2011] can be performed, and then it becomes another semantifying problem between entities problem.

In this paper we focus on the problem of semantifying the first type of hyperlinks since other kinds of links can finally be deduced to this problem.

We argue that the context of an entity in an article is informative and deserves a higher occurrence and can produce fresh and latest relation of an entity, which can be useful in updating the KB [reverb]. In our approach the relations are not necessarily to exist, however the cluster of the relation [relation clustering] it belongs to should be conceptually right. For instance, the relationship `artist of` will be correct in the tuple of (Leonardo Da Vinci, `artist of`, Mona Lisa) [(Human, `artist of`, Painting)] and will be never correct in the tuple of (Automobile, `artist of`, Paint-

ing) [e.g. (BMW i8, `artist of`, Mona Lisa )]. With rich context and large knowledge base, we can easily derive the fresh context relations and the concept of each entity,

On the other hand, we consider the co-occurrence of the 2 entities, based on the assumption of important relationship will be observed in various of documents.

For these reasons, we propose a relation explanation method leveraging the concept and co-occurrence of an entity, to explain the relation between the target entity and the related entity, thus semantifying the hyperlink.

### 2. RELATED WORKS

#### 2.1 Link entities to Database

Linking entities to database, especially to Wikipedia, has been widely studied. Entity linking to Wikipedia [10, 9, 5] exploits Wikipedia as thesaurus and link web documents to it. In our work, instead of linking entities to the corresponding one in KB, we extract the target entity [1] and explain the semantic relation of the entity towards target entity.

#### 2.2 Relation Discovery

to be read [12, 11, 7]

Recently, different efforts are devoted to relation Discovery [3, 13, 8] are studied on graph based approaches and text based approaches [6] [reverb]. However, these approaches will be limited largely by the incompleteness of Knowledge Base, and cannot discover new type of relations. In our work, we focus on semantic relations of entity by leveraging the concepts of entity extracted from knowledge bases to link entities with a probability.

#### 2.3 Information Extraction

Attribute acquisition methods Among domain-dependent approaches, we can mention approaches that focus on products. In this domain, attributes have been used to improve product search and recommendation [18, 22], but also to enable data mining [27]

Attribute retrieval provides another granularity in Web search. This can interest communities that propose a more focused access to information or communities that envision aggregating pieces of information such as aggregated search [19, 15]. Wong et al. [27] combine tags and textual features in a Conditional Random Fields model to learn attribute extraction rules, but they need a seed of relevant documents manually fed.

#### 2.4 Short Text Conceptualization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

### 3. PROBABILITY RECALCULATION AFTER CONCEPTUALIZATION

#### 3.1 Problem Statement

##### intro about probase

Given an entity  $e$ , from Probase, we can acquire its concepts' set  $C$  and for each  $c_i \in C$ , the frequency  $n(c_i, e)$  can be accordingly derived, which means how many times the  $e$  is a  $c_i$  pattern can be observed from the original corpus. we can derive  $P(c_i|e)$ , where  $c_i \in C_{probase}$ ,

$$P(c_i|e) = \frac{n(c_i, e)}{n(e)}$$

However, the concepts here has various forms as illustrated in Example 1. For our task, we only need relatively general concepts (a.k.a. head concepts).

EXAMPLE 1 (VARIOUS FORMS OF CONCEPTS). *Take the entity Mona Lisa as example, its concepts includes painting, famous painting, world's most famous painting, with corresponding frequency 33, 8, 1*

We divide the  $C_{Probase}$  into 2 parts,  $C_{simple}$  and  $C_l$ , where  $C_{simple}$  only contains one word and  $C_l$  are the rest.  $C_{simple}$  are generated by the head modifier detection. The problem here is to recalculate the probability  $P(c_h|e)$  where  $c_h \in C_{simple}$ , literally, we should contribute all the counts of  $C_l$  to  $C_{simple}$ .

##### Why head concepts?

The relationship between entities are determined by the simple concepts. For example, the **founder** relationship between **Apple Inc.** and **Steve Jobs** are determined by the head concepts they possessed (e.g. **company** and **entrepreneur**, regardless of the modifiers such as **technology** in the concept **technology company**.) The number of entities can be very large, but the number of top concepts and the relationship between them are limited, literally, we can find all the possible relationship between concepts instead of store all the long-tailed entities and their relations, which indicates the rationality of doing conceptualization. The reason why we do head modifier detection instead of directly using isA edge in probase, is that even if the long concept  $c_l$  has an isA edge towards a certain concept  $c_h'$ , it still sometimes not include the head concept of the long concept which is very plausible as is demonstrated in Example 2

EXAMPLE 2 (HEAD CONCEPTS VS ORIGINAL CONCEPTS). *Take famous painting as example. Its original concepts are image, treasure, which are reasonable but not plausible, since their occurrence are 2 and 1 respectively. However, the most plausible concept painting is not among the concepts.*

##### The main steps.

Given an entity  $e$  from Probase, we can get its concepts from probase. First we do head modifier detection based on syntax[], since the concepts in Probase all follows English grammar, this approach already produces a good result. Next, we recalculate the probability of  $P(c_h|e)$  by aggregating the contribution from  $c_l$ . The essentiality of doing

so is illustrated in Example 3. Finally, we provide a method to take the original isA relation from Probase into consideration.

EXAMPLE 3 (ESSENTIALITY OF AGGREGATION). *steve jobs The concept well-known name has four occurrences however name has only 2. There are other modifiers for the same head, so that the typicality of the head will be largely underestimated.*

#### 3.2 Baseline

After head modifier detection, we have a set of  $c_h \in C_{simple}$ , among all the  $c_{l_j} \in C_l$ , there are 2 cases in the probase determined by whether the  $c_{l_j}$  has an isA edge towards  $c_h$  or not. The intuition of doing so is illustrated in the Example 4:

EXAMPLE 4 (CONTRIBUTING LONG CONCEPTS). *Assume that Mona Lisa is a painting and Mona Lisa is a famous painting are observed respectively 33 times and 8 times from different documents, we will get the knowledge that Mona Lisa is a painting occurs 41 times instead of 33 times.*

Hence, the most straight forward approach is to contribute the corresponding long concepts to the simple ones as follows:

$$\hat{P}(c_h|e) = \frac{n(c_h, e) + \sum_{f_{HM}(c_l)=c_h} n(c_l, e)}{\sum n(c_h^*, e)}$$

where  $f_{HM}()$  is a function that takes a long concept and produce a head concept.

#### 3.3 Combined Model with Original IsA

When obtaining  $\hat{P}(c_h|e)$ , we are in fact judging the typicality of a concept. In this section, we take the original Probase IsA relation into consideration. In Example. 5, we can observe some reasonable results produced by the Probase isA relationship.

EXAMPLE 5 (REASONABLE IS A RELATION). *There exists several original IsA concepts of the long concepts that are also reasonable. For example topaz (a kind of yellow gemstone) has the concept precious stones, and precious stones has an edge towards material which is reasonable.*

Based on the how to treat  $c_l$ , we have the following 2 cases:

**$c_l$  appear as an concept** In this case the counts that its entities produced should be take into consideration into  $c_h$ , deriving the following Case A.

**$c_l$  appear as an instance** In this case  $c_l$  is observed from the corpus as an instance at the left side of an isA sentence, it should be treated the same as other entities  $e$ , deriving the following Case B.

Therefore, to calculate  $P(c_h|e)$ , there are three cases.:

**Case A.1**  $e \xrightarrow{isA} c_h$  The entity has an isA edge towards one or more simple concept, which gives the original  $P_{org}(c_h|e) =$

**Case A.2**  $e \xrightarrow{isA} c_l \xrightarrow{Head} c_h$  The solid edge here refers to the isA relationship in *Probase* and the dashed one refers to the edge generated by head modifier detection. Example 6 pointed out that there won't be necessarily an isA edge from **famous painting**( $c_l$ ) to **painting**( $c_h$ ), however  $c_l$  is obviously a hyponym of  $c_h$ . In this case, we have to re-calculate the  $P(c_h|c_l)$ . In the original probase approach, we use Eq. 1 to calculate the probability.

$$P(c_h|c_l) = \frac{n(c_h, c_l)}{\sum n(c_h^*, c_l)} \quad (1)$$

However,  $n(c_h, c_l)$  is lower than expected due to the reason demonstrated in Example. 6. Therefore, we alternatively utilize the  $\sum e^*, n(c_l)$  as the occurrence of  $c_l$ , following the assumption that *whether  $c_l$  is typical towards its  $c_h$  is independent from*

$$P_{head}(c_h|e) = \frac{\sum_{c_h=f_{HM}(c_l^*)} n(c_l^*, e)}{n(e)}$$

**Case B**  $e \xrightarrow{isA} c_l \xrightarrow{isA} c_h$  In this case, we need to calculate the following equation

$$P(c_h|e) = \sum_{c_l^* \in C_l} P(c_h|c_l^*, e) \times P(c_l^*|e)$$

, where  $P(c_l^*|e)$  can be obtained from *Probase* and

$$P(c_h|c_l, e) = \frac{n(c_h, c_l, e)}{n(c_h, e)} \quad (2)$$

We assume that the occurrence of  $e$  does not affect  $P(c_h|c_l)$  equivalently speaking,  $P(c_h|c_l)$  is independent from  $e$ , thus Eq. 2 can be simplified

$$P(c_h|c_l, e) = P_{probase}(c_h|c_l) = \frac{n(c_h, c_l)}{n(c_h)}$$

which can be obtained from *Probase*.

**EXAMPLE 6 (WHY AREN'T HEAD RELATIONSHIP OBSERVED).** *There are less chance of occurring Famous painting is a painting in the corpus, since human takes it for granted and will seldom express it in such a way, so that there won't be necessarily an isA edge from famous painting to painting in the KB, while we insist it is necessary.*

Considering Case A.1 and Case A.2 we get the baseline. When calculating typicality, we should consider the both cases of  $c_l$ , thus combining Case A and B through a linear combination.

Finally  $P(c_h|e)$  is calculated using the following equation:

$$P(c_h|e) = \alpha \hat{P}(c_h|e) + (1 - \alpha) \sum_{c_l^* \in C_l} [P(c_h|c_l^*)] \times P(c_l^*|e) \quad (3)$$

$$P(c_h|e) = \sum_{c_l^* \in C_l} [P(c_h|c_l^*)] \times P(c_l^*|e) \quad (4)$$

We consider only 2 layers of isA relationship for 2 reasons. The first one is that more layers will lead to noisy concepts such as **issue**, **factor**, **element**, which are concepts for almost everything, Secondly, discussing the transitive relation between concepts is beyond the scope of this paper.

## 4. FIND ALIAS FOR ATTRIBUTES

For a pair (Sherlock holmes, United Kingdom), country is a merely-ok attribute, on the contrary, residence, deathPlace are better since they are more specific and more seemingly plausible to be an attribute.

### 4.1 Calculating $P(a|(e_1, e_2))$

Given an entity pair  $(e_1, e_2)$ , we want to find the best attribute to describe the semantic meaning between them, which can be formalized as calculating:

$$P(a|(e_1, e_2))$$

for each attribute given the entity pair.

For any  $(entity_1, attribute, entity_2)$  tuple, later denoted as  $(e_1, a, e_2)$ , where  $e_1$  and  $e_2$  are also referred to as **domain** and **range** of the attribute. We can conceptualize  $e_1$  and  $e_2$  using the method in section 3, and get a set of concept  $C_1, C_2$ , accompanied with a set of probabilities  $P(\gamma_1|e_{1i}), P(\gamma_2|e_{2j})$ , where  $\gamma_1 \in C_1, \gamma_2 \in C_2$ .

To construct the Entity Attribute Graph, we only need topK concepts to form  $(\gamma_1, \gamma_2)$  pair, K through **case study** is around 5, so we here set  $K=10$ .

Thus for any attribute  $a$ , given a pair of entity  $(e_{1i}, e_{2j})$ , we can define: **should i use joint ratio here?**

$$\begin{aligned} P_{(e_{1i}, e_{2j})}((\gamma_1, \gamma_2)|a) &= P_{before}(\gamma_1|a) \times P_{after}(\gamma_2|a) \\ &= P(\gamma_1|e_{1i})P(e_{1i}|a) \times P(\gamma_2|e_{2j})P(e_{2j}|a) \end{aligned} \quad (5)$$

where we use  $P_{(e_{1i}, e_{2j})}((\gamma_1, \gamma_2)|a)$  to denote observing a single pair  $(e_{1i}, e_{2j})$ , how likely is a combination of  $(\gamma_1, a, \gamma_2)$  to occur.

Consequently,

$$P((\gamma_1, \gamma_2)|a) = \sum_{e_{1i} \in E_1, e_{2j} \in E_2} P_{(e_{1i}, e_{2j})}((\gamma_1, \gamma_2)|a) \quad (6)$$

where  $E_1, E_2$  denoting the whole set of domain entity and range entity, The  $P(e_{1i}|a)$  and  $P(e_{2j}|a)$  here has only 2 values 1 and 0, depending on whether  $e_1$  occurs before  $a$  or  $e_2$  occurs after  $a$ . Apparently, only  $(e_{1i}, a, e_{2j})$  occurs will give the equation a non-zero value, therefore, Eq. 6 is finally equal to Eq. 7.

$$\begin{aligned} P((\gamma_1, \gamma_2)|a) &= \sum_{(e_{1i}, a, e_{2j}) \in KB} P_{(e_{1i}, e_{2j})}((\gamma_1, \gamma_2)|a) \\ &= \sum_{(e_{1i}, a, e_{2j}) \in KB} P(\gamma_1|e_{1i}) \times P(\gamma_2|e_{2j}) \end{aligned} \quad (7)$$

The process of calculating is demonstrated in Example. 7

**EXAMPLE 7 (CALCULATING  $P((\gamma_1, \gamma_2)|a)$ ).** *As illustrated in Fig. 2, the process of calculating  $P((\gamma_1, \gamma_2)|a)$  is as follows*

**insert a graph**

To Construct the Entity Attribute Graph, we calculate  $P((\gamma_1, \gamma_2)|a)$  for each attribute.

Note that we only consider the attributes whose range is an entity, and ignore those numerical values or date-and-time values such as (*MonaLisa*, *Year*, 1503).

For each  $(\gamma_1, a, \gamma_2)$  tuple, we can calculate  $P((\gamma_1, \gamma_2)|a)$  for each

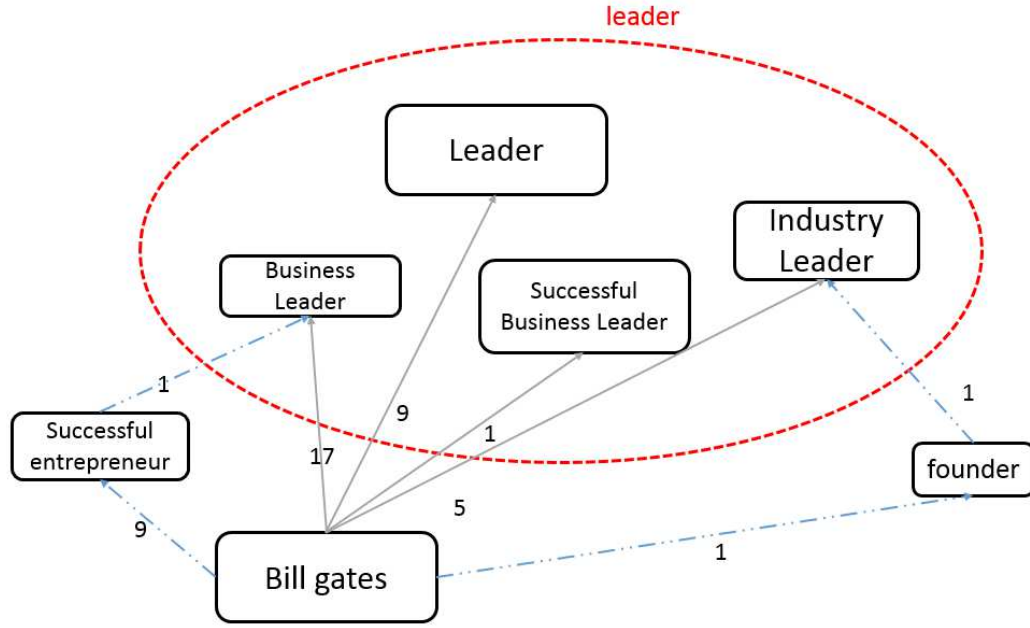


Figure 1: calculating  $P(c_h|BillGates)$

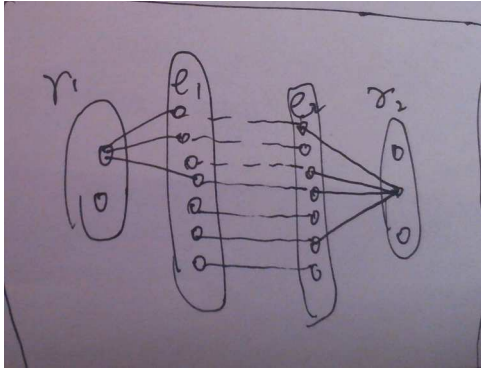


Figure 2: Calculating  $P((\gamma_1, \gamma_2)|a)$  for

## 4.2 For multiple hops

### 4.2.1 Entity Attribute Graph Construction

So far, we have tackled with the relations and generated the edges in the relationship graph. This problem is similar to **hierarchy ranking problems in a directed graph** [4]. Originally, it was a minimum feedback arc set problem on a weighted network which is a classic NP-hard problem [2]. A few approaches [14] have been proposed on unweighted directed graphs, for weighted graphs, the extended agony[hierarchies in directed network(unpublished kdd15)] algorithm can be utilized to generate hierarchy results in this approach the K (number of hierarchies) is fixed, maybe we can make it adaptive to data here?

In this section, we first formulate the problem of finding semantic link into a maximum flow problem on the concept network with multiple-sources and multiple-sinks, and then, we cut out the subgraph and perform **improved agony** to

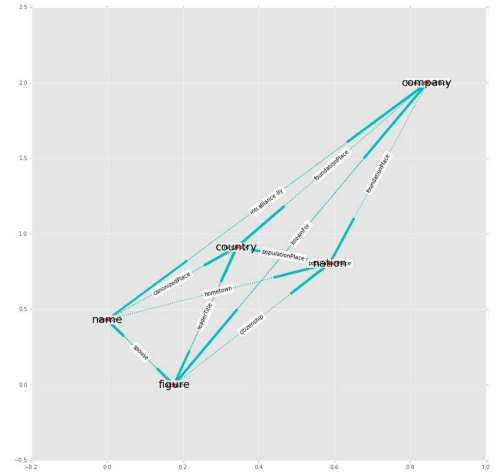


Figure 3: Subgraph of Entity Attribute Graph

derive the concept of the middle entities. Last we use co-occurrence to verify the validness of the relation.

### 4.2.2 Improved agony

## 4.3 Find the best alias

We then Use an argmax model use KL divergence? to minimize  $D_{KL}$  to solve the problem.

Given  $(e_1, e_2)$ , our goal is to find the best attribute for it. We denote it as:

$$\arg \max P((e_1, e_2)|a)$$

where

$$P((e_1, e_2)|a) =$$

## 5. EXPERIMENT

### 5.1 Experiment Setup

We

### 5.2 Evaluation

In

### 5.3 Head Concept Vs Original Concept

### 5.4 Find alias

#### 5.4.1 compare

Compare  $P((\gamma_{1i}, \gamma_{2i}|a))P(\gamma_{1i}|a) \times P(\gamma_{2i}|a)$

#### 5.4.2 Sense Disambiguation

We can solve the problem of sense disambiguation problem well by applying this method since there are many entities belongs to the same concept and we only consider topK  $(\gamma_1, \gamma_2)$  pairs that has high typicality  $P((\gamma_1, \gamma_2)|a)$ , so that the weird  $(\gamma_1, \gamma_2)$  patterns as manifest in Example. 8 can be easily filtered.

cut the figure smaller



Figure 4:  $(\gamma_1, \gamma_2)$  plot for attribute Manufacturer

EXAMPLE 8 (SENSE DISAMBIGUATION). Consider the following  $(e_1, a, e_2)$  tuple (iphone, manufacturer, apple). Suppose it is our query, where apple's sense can either be a kind of fruit or a company. Fig. 4 is a heatmap for all the concepts pairs  $(\gamma_1, \gamma_2)$  of attributes manufacturer. The horizontal axis represents the  $e_1$  and the vertical axis stands for  $e_2$ . The darker the blue is, the higher typicality it will be. In Fig. 4, We can observe that the top concepts of  $e_2$  in the heatmap are company, manufacturer, ... and top 10 pairs also does not include fruit. The intuition for this is that there exists thousands of  $(e_1, a, e_2)$  tuple such as (BMW\_Z4, manufacturer, BMW), (PlayStation\_4, manufacturer, Sony)

other than (iphone, manufacturer, apple) tuple, which results in a reasonable distribution.

## 6. CONCLUSION

## 7. REFERENCES

- [1] N. Dalvi, R. Kumar, and M. Soliman. Automatic wrappers for large scale web extraction. *Proceedings of the VLDB Endowment*, 4(4):219–230, 2011.
- [2] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005.
- [3] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon. Rex: explaining relationships between entity pairs. *Proceedings of the VLDB Endowment*, 5(3):241–252, 2011.
- [4] M. Gupte, P. Shankar, J. Li, S. Muthukrishnan, and L. Iftode. Finding hierarchy in directed online social networks. In *Proceedings of the 20th international conference on World wide web*, pages 557–566. ACM, 2011.
- [5] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM, 2011.
- [6] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415. Association for Computational Linguistics, 2004.
- [7] N. Konstantinova. Review of relation extraction methods: What is new out there? In *Analysis of Images, Social Networks and Texts*, pages 15–28. Springer, 2014.
- [8] G. Luo, C. Tang, and Y.-I. Tian. Answering relationship queries on the web. In *Proceedings of the 16th international conference on World Wide Web*, pages 561–570. ACM, 2007.
- [9] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.
- [10] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.
- [11] N. Nakashole, G. Weikum, and F. Suchanek. Discovering and exploring relations on the web. *Proceedings of the VLDB Endowment*, 5(12):1982–1985, 2012.
- [12] N. Nakashole, G. Weikum, and F. Suchanek. Discovering semantic relations from the web and organizing them with patty. *ACM SIGMOD Record*, 42(2):29–34, 2013.
- [13] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–632. ACM, 2010.



**Table 1: Rerank comparison**

Entity	aggregated Top 10			original head concepts		
shanghai	agg head	agg count	agg prob	org head	org count	org prob
	city	1311	0.829222	city	644	0.407337
	region	46	0.029096	region	27	0.017078
	area	42	0.026565	metropolis	23	0.014548
	metropolis	26	0.016445	megacities	15	0.009488
	port	20	0.01265	market	15	0.009488
	market	19	0.012018	location	15	0.009488
	centre	18	0.011385	port	9	0.005693
	location	17	0.010753	locality	6	0.003795
	megacities	15	0.009488	locale	5	0.003163
bill gates	center	11	0.006958	seaport	4	0.00253
	leader	46	0.140244	billionaire	37	0.112805
	billionaire	44	0.134146	entrepreneur	28	0.085366
	entrepreneur	41	0.125	philanthropist	23	0.070122
	philanthropist	30	0.091463	celebrity	15	0.045732
	celebrity	20	0.060976	leader	9	0.027439
	person	16	0.04878	innovator	6	0.018293
	figure	11	0.033537	personality	5	0.015244
	innovator	8	0.02439	expert	5	0.015244
	luminary	8	0.02439	folks	4	0.012195
samsung	individual	7	0.021341	icon	4	0.012195
	company	1030	0.376875	company	816	0.298573
	brand	829	0.30333	brand	561	0.205269
	manufacturer	238	0.087084	client	42	0.015368
	maker	112	0.040981	firm	39	0.01427
	player	60	0.021954	rival	38	0.013904
	phone	60	0.021954	player	33	0.012075
	giant	51	0.018661	phone	30	0.010977
	firm	49	0.017929	conglomerate	19	0.006952
	name	49	0.017929	corporation	19	0.006952
mona lisa	conglomerate	42	0.015368	partner	12	0.004391
	painting	56	0.4	painting	33	0.235714
	masterpiece	21	0.15	masterpiece	16	0.114286
	work	20	0.142857	work	10	0.071429
	film	6	0.042857	film	5	0.035714
	image	5	0.035714	image	3	0.021429
	artwork	4	0.028571	picture	3	0.021429
	portrait	4	0.028571	treasure	2	0.014286
	piece	4	0.028571	song	2	0.014286
	picture	3	0.021429	icon	2	0.014286
	figure	3	0.021429	artwork	1	0.007143

- [14] N. Tatti. Faster way to agony. In *Machine Learning and Knowledge Discovery in Databases*, pages 163–178. Springer, 2014.