Wikipedia-Based Distributional Semantics for Entity Relatedness

Nitish Aggarwal and **Paul Buitelaar**
Insight Centre for Data Analytics
National University of Ireland
Galway, Ireland
firstname.lastname@insight-centre.org

Abstract

Wikipedia provides an enormous amount of background knowledge to reason about the semantic relatedness between two entities. We propose Wikipedia-based Distributional Semantics for Entity Relatedness (DiSER), which represents the semantics of an entity by its distribution in the high dimensional concept space derived from Wikipedia. DiSER measures the semantic relatedness between two entities by quantifying the distance between the corresponding high-dimensional vectors. DiSER builds the model by taking the annotated entities only, therefore it improves over existing approaches, which do not distinguish between an entity and its surface form. We evaluate the approach on a benchmark that contains the relative entity relatedness scores for 420 entity pairs. Our approach improves the accuracy by 12% on state of the art methods for computing entity relatedness. We also show an evaluation of DiSER in the Entity Disambiguation task on a dataset of 50 sentences with highly ambiguous entity mentions. It shows an improvement of 10% in precision over the best performing methods.

In order to provide the resource that can be used to find out all the related entities for a given entity, a graph is constructed, where the nodes represent Wikipedia entities and the relatedness scores are reflected by the edges. Wikipedia contains more than 4.1 millions entities, which required efficient computation of the relatedness scores between the corresponding 17 trillions of entity-pairs.

Introduction

Entities like persons, locations, organizations etc. are the key features to define the semantics of natural language text. Significance of measuring relatedness between entities has been shown in various tasks which deal with information retrieval (IR), natural language processing (NLP), text analysis or other related fields. For instance, entity disambiguation (Hoffart et al. 2011; 2012; Kulkarni et al. 2009) mainly relies on the quality of entity relatedness measures to jointly map the surface forms to their defining entities registered in knowledge bases. The measure is also very useful to obtain the related entities in query expansion (Aggarwal and Buitelaar 2012; Pantel and Fuxman 2011), knowledge base population (Ji et al. 2010), semantic search (Demartini et al. 2010), and other similar tasks.

Reasoning about the semantic relatedness of "apple" and "next" requires an immense amount of world knowledge about the concepts represented by these two surface forms. The semantic meaning of "apple" may refer to a fruit, person's surname or a company. Similarly, "next" refers to more than 20 different entities on Wikipedia but the most common meaning of "next" that comes first in mind is "succeeding item". It is hard to assess the relatedness between "apple" and "next" as they are highly ambiguous. However, if we are given that both "apple"[1] and "next"[2] are software companies founded by Steve Jobs, we can easily judge their appropriate relatedness.

Semantic meaning of an entity can be inferred from its distribution in a high dimensional space of concepts derived from Wikipedia as it is a constantly growing encyclopedia, containing world knowledge about millions of entities. The usage of an entity in large contextual space can be utilized to build a distributional vector (Harris 1954). Therefore, we can build a high dimensional vector over the Wikipedia concepts by taking every concept as a dimension. The associativity weight of an entity with the concept can be taken as the magnitude of the corresponding dimension in the vector. To obtain the semantic relatedness between two entities, we can simply quantify the distance between their distributional vectors. Some of the existing approaches (Gabrilovich and Markovitch 2007; Landauer, Foltz, and Laham 1998) calculate the semantic relatedness between two natural language texts by using the distributional vectors. However, they are limited to perform only at surface level of the entities rather than on their registered definitions in knowledge repositories. Due to this limitation, these methods can produce an ambiguous distributional vector for ambiguous surface forms like "apple" or "next". For instance, if we retrieve the most associated Wikipedia concepts for the term "next", we will

[1] http://en.wikipedia.org/wiki/Apple_Inc.

[2] http://en.wikipedia.org/wiki/NeXT

get concepts like *linked list, railway station, train schedule*. However, if we retrieve the concepts which only contain the entity "NeXT" as annotated Wikipedia link, we will get concepts like *Music Kit, NeXT, NeXT Computer*.

In this paper, we present Wikipedia-based Distributional Semantics for Entity Relatedness (DiSER), which represents the semantics of an entity by a high dimensional vector. DiSER builds this semantic vector over Wikipedia concepts by taking the annotated entities only. Therefore, it eliminates a limitation in the existing approaches (Gabrilovich and Markovitch 2007; Landauer, Foltz, and Laham 1998) which do not differentiate between "apple" fruit and "Apple" company. Since DiSER can generate the vectors only for those entities which appear on Wikipedia, we also propose an alternative approach called Context-DiSER that builds a DiSER vector by taking additional resources such as music portal, personal websites, blogs or social network websites into account for retrieving the context. Context-DiSER eliminates the dependency of having predefined direct interlinkage between the given entities, which is required for existing approaches (Witten and Milne 2008; Ponzetto and Strube 2007). Hoffart et al. (2012) utilize the context of entities to calculate semantic relatedness by taking the overlap between corresponding contexts, however, their method does not semantically interpret the context. Our goal is to develop an entity relatedness measure which overcomes these limitations and improves over the existing algorithms. We evaluate our approach in two different tasks: ranking related entities and entity disambiguation.

Obtaining a ranked list of related entities is required by various tasks such as query suggestion in web search (Blanco et al. 2013; Yu et al. 2014), query expansion (Pantel and Fuxman 2011), and recommendation systems. Many users search for a particular entity and like to explore about related entities. These related entities provides an opportunity to the users to extend their knowledge. Related entities can be obtained from knowledge bases such as DBpedia or Freebase by retrieving the directly connected entities. However, most of the popular entities have more than 1,000 directly connected entities, and these knowledge bases mainly cover some specific types of relations. For instance, "Steve Jobs" and "Steve Wozniak" are not directly connected in DBpedia graph. Therefore, we need to find related entities beyond the relations defined in Knowledge base graphs. Further, these related entities required a ranking method to select the most related entities. Similarly, recommendation systems require a method to select the most related items to some given items which are liked by a user. Therefore, we build a entity relatedness graph (EnRG)[3], where every node of the graph represents a Wikipedia article (entity) and their relatedness scores are reflected by the edges between them. We consider every Wikipedia article as an entity. Wikipedia contains more than 4.1 millions entities. Therefore, to build this graph, we need to calculate the relatedness scores between 16.8 trillions (4.1 millions x 4.1 millions) of entity-pairs. The graph provides a ranked list of related entities to a given entity. Every Wikipedia article has a corresponding DBpedia page that provides further exploration of different relations of the entity with others. DBpedia defines the rdf:type of every Wikipedia entity, which allows us to get the ranked list of a particular type. The rdf:type information enables us to retrieve different aspects of the related entities by grouping them in related people, companies, locations and other types of entities. For instance, if we are interested to find the related entities of "Apple Inc.", we may obtain "Steve Jobs", "Steve Wozniak" and "Tim Cook" as top related people; "NeXT", "Pixar" and "Motorola" as top related companies; and other types of entities such as "iPod", "OS X", "iPhone", "iPad" and many more.

Related Work

Text Relatedness

In recent years, there have been a variety of efforts to develop semantic relatedness measures for natural language texts. Classical approaches assess the relatedness scores by comparing the texts as bags of words in a vector space model (Manning, Raghavan, and Schütze 2008). Most of these approaches make use of manually constructed lexical resources like WordNet to calculate the relatedness between words. For instance, (Hirst and St-Onge 1998) and (Wu and Palmer 1994) utilize the edges that define taxonomic relations between words; Banerjee and Pedersen (2002) compute the scores by obtaining the overlap between glosses associated with the words; and some of the other approaches (Resnik 1995; Pirró and Euzenat 2010) use corpus evidence with the taxonomic structure of WordNet. These approaches are limited to perform only for lexical entries and do not work with named entities.

Corpus-based methods such as Latent Semantic Analysis (LSA) (Landauer, Foltz, and Laham 1998), Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) and Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2007) employ statistical models to build the semantic profile of a word. LSA and LDA generate unsupervised topics from a textual corpus, and represent the semantics of a word by its distribution over these topics. However, ESA directly uses supervised topics such as Wikipedia concepts that are built manually. ESA generates the vector for a given word over Wikipedia concepts in which it occurs. Several variations (Hassan and Mihalcea 2011; Polajnar et al. 2013; Aggarwal, Asooja, and Buitelaar 2014) have been proposed to improve the ESA performance. Polajnar et al. (2013) perform query expansion in creating the vectors, and Hassan and Mihalcea (2011) take entity co-occurrences into account. These corpus based approaches overcome the dependency on predefined lexical resources, but they are unable to handle ambiguous terms such as "apple" and "next" as entities.

Entity Relatedness

Wikipedia and its derived knowledge bases like DBpedia (Auer et al. 2007), YAGO (Suchanek, Kasneci, and Weikum

[3] http://server1.nlp.insight-centre.org:8080/enrg/

2007) and FreeBase (Bollacker et al. 2008) provide an immense amount of information about millions of entities. The advent of this knowledge about persons, locations, products, events etc. introduces numerous opportunities to develop entity relatedness measures. Strube and Ponzetto (2006) proposed WikiRelate that exploits the Wikipedia link structure to compute the relatedness between Wikipedia concepts. WikiRelate counts the edges between two concepts and takes the depth of a concept in the Wikipedia category structure into account. Ponzetto and Strube (2007) adapted WordNet-based measures to Wikipedia for obtaining the advantages of the constantly growing vocabulary of Wikipedia. Witten and Milne (2008) applied the Google distance metric (Cilibrasi and Vitanyi 2007) on incoming links in Wikipedia. These approaches perform only for the entities which appear on Wikipedia. KORE (Hoffart et al. 2012) eliminates this issue by computing the relatedness scores between the context of two entities. It observes the partial overlaps between the concepts (key-phrases) appearing in the context of the given two entities. KORE improves over other existing algorithms by taking entity context into account. However, it considers only the surface forms of the concepts appearing in the context and does not utilize their background knowledge.

Entity Recommendation

Entity recommendation can be defined by finding a ranked list of related entities for a given entity. Blanco et al. (2013) introduced Spark that links a user search query to an entity in a knowledge base and provides a ranked list of related entities for further exploration. Spark uses different features from several resources such as Flickr, Yahoo query logs and the Yahoo Knowledge graph. Spark tunes the parameters by using training data. Similarly, Yu et al. (2014) proposed personalized entity recommendation which uses several features extracted from user click logs provided through Bing search. Search engine specific datasets are not publicly available. Furthermore, it is expensive to create a training dataset to combine several features by using machine learning methods.

Approach

Computing Entity Relatedness

We developed an approach called Wikipedia-based Distributional Semantics for Entity Relatedness (DiSER), which builds the semantic profile of an entity by using the high dimensional concept space derived from Wikipedia. DiSER generates a high dimensional vector by taking every Wikipedia concept as dimension, and the associativity weight of an entity with the concept as the magnitude of the corresponding dimension. To measure the semantic relatedness between two entities, we simply calculate the cosine score between their corresponding DiSER vectors.

We retrieve a list of relevant Wikipedia concepts and rank them according to their relevance scores with the given entity. DiSER considers only human annotated entities in Wikipedia, thus keeping all the canonical entities that appear with hyperlinks in Wikipedia articles. The tf-idf weight of an entity with every Wikipedia article is calculated and used to build the corresponding semantic profile, which is represented by the retrieved Wikipedia concepts sorted by their tf-idf scores. For instance, for an entity *e*, DiSER builds a semantic vector *v*, where $v = \sum_{i=0}^{N} a_i * c_i$ and c_i is i^{th} concept in the Wikipedia concept space, and a_i is the tf-idf weight of the entity *e* with the concept c_i. Here, N represents the total number of Wikipedia concepts.

DiSER is able to capture the semantic meaning of entities like “Apple” and “NeXT”. Therefore, it can improve over existing algorithms that build the distributional vector by taking only the surface forms into account. For instance, ESA generates the distributional vector of a term by calculating its tf-idf weight with all the Wikipedia articles. Table 1 shows the 10 most associated concepts obtained by ESA and DiSER. It illustrates that existing systems can not handle ambiguous terms as they generate the vector by considering the surface form of an entity. Manual analysis of the vectors generated by ESA and DiSER reveals that all the concepts retrieved by DiSER are relevant to the entity “NeXT”. However, ESA did not get any relevant concept as it is more biased towards the global meaning of the given term. Table 1 shows that ESA retrieved all concepts for the word “next” with the semantic meaning “succeeding item”. Therefore, it retrieved concepts which contain phrases like “next item in linked list” or “next train”.

Since the model relies on the notion “entities that co-occur together more often, tend to be more related”, it is limited to perform only for the entities which appear and co-occur in Wikipedia. We resolve this issue by generating the DiSER vector of entities appearing in the context of a given entity. For instance, Wikipedia does not have a page for “Matt Lasky” who is a Hollywood actor. Therefore, DiSER cannot generate the semantic profile for “Matt Lasky”. However, the IMDB page[4] and his website[5] define him with the entities appearing on Wikipedia such as “Hollywood”, “Actor”, “Pirates of the Caribbean” and “Princess of Mars”. These entities can be identified by using any ex-

	NeXT ESA vector	**NeXT DiSER vector**
1.	Doubly linked list	Music Kit
2.	Gare de Rennes	NeXTSTEP
3.	Brugge railway station	NeXT Laser Printer
4.	Gare d’Avignon Centre	NeXT Computer
5.	Gare de Toulon	Shelf (computing)
6.	Szczecin Glowny railway station	RichPage
7.	Gare de Clermont Ferrand	ISPW
8.	Leipzig Central Station	Nancy R. Heinen
9.	Brussels North railway station	Enterprise Objects Framework
10.	Gare de Strasbourg	Lotus Improv

Table 1: 10 most relevant Wikipedia concepts for entity “NeXT” by ESA and DiSER

[4] http://www.imdb.com/name/nm2359377/

[5] http://mattlasky.nowcasting.com/

isting entity linking tools such as Alchemy[6] or Zemanta[7]. With these Wikipedia concepts, we can build the semantic profile of "Matt Lasky". We describe above the method to build a DiSER vector for one entity. Similarly, we can build it for more, by treating them as a bag of entities. Let $E = \{e_1, e_2, e_3, ...e_n\}$, where E is a set of entities. We generate a vector V, where $V = \sum_{e_k \epsilon E} v_k$ and $v_k = \sum_{i=0}^{N} a_i * c_i$. v_k represents the DiSER vector as explained above.

Implementation

We implemented our approach by using the snapshot of English Wikipedia from Oct 1^{st}, 2013. This snapshot consists of 13,872,614 articles, in which 5,659,383 are Wikipedia redirects. Wikipedia redirects are those articles, which do not contain any content, and just link to the article or the section of the article that defines a similar or related Wikipedia concept. For example, U.S.A. redirects to United States. Wikipedia also contains namespace pages, which are reserved by MediaWiki to describe the Wikipedia projects like Wikipedia help and File pages. We filtered out all the namespace pages by using the articles title as they have specific namespace patterns. There were 3,571,206 namespace pages in this snapshot. We also removed all those articles which contain less than two human annotated entities; such articles would not provide any co-occurrence information for the entities. Finally, we obtained 4,102,442 articles after removing all the Wikipedia redirects, namespaces and short articles.

In order to use the annotated entities only for generating the DiSER vectors, it is required to retain only those entities which have manually defined links provided by Wikipedia volunteers. However, the volunteers may not create a link for every surface form appearing in the article content. For instance, "apple" occurs 213 times in the "Steve Jobs" Wikipedia page, but only 7 out of these 213 are linked to the "Apple Inc." Wikipedia page. This term frequency of "apple" is calculated without considering the partial matches, for example, we do not count if "apple" appears as a substring of any annotated entities like "Apple Store" or "Apple Lisa". Since we measure the associativity of an entity with the article by computing its tf-idf weight, this difference in term frequencies may have a major effect on the semantic profile interpreter. Therefore, to obtain the actual term frequency of every entity, we apply the "one sense per discourse" heuristic (Gale, Church, and Yarowsky 1992). According to which, a term tends to have the same meaning in the same discourse. We annotated every additional unannotated occurrence of a term with the hyperlink appearing several times for the same term in the discourse.

Since DiSER builds the distributional vector for the given entity by calculating its tf-idf scores with every article, it may take a very long time to process 4.1 millions articles. Therefore, we built an inverted index of 4.1 million preprocessed articles by using Lucene[8]. As this indexing is a one time process to build the DiSER vector, we can calculate entity relatedness for several thousand pairs within a second.

Evaluation in Ranking Related Entities

Dataset

In order to compare our approach against existing entity relatedness measures, we performed our experiments on the same gold standard dataset KORE (Hoffart et al. 2012) that has been used by state of the art methods.

The KORE dataset consists of 21 seed entities selected from the YAGO knowledge base. Every seed entity has a ranked list of 20 related entities. The seed entities are selected from 4 different domains: IT companies, Hollywood celebrities, video games, and television series. The dataset consists of very popular entities from these 4 domains like Google and IBM as IT companies, Brad Pitt and Angelina Jolie as Hollywood celebrities, Max Payne and Quake as video games, and Futurama and The Sopranos as television series. 20 entity candidates are selected for every seed entity. It is very difficult to judge an absolute relatedness score between two entities, and most of the applications require a ranked list of entities. Therefore, these candidates were given to human evaluators on crowdsourcing to give the relative comparison between two candidates against the corresponding seed entity. For instance, human evaluators provide their judgement if "Mark Zuckerberg" is more related to "Facebook" than "Sean Parker". With the answers of these types of binary questions, a ranked list was prepared for every seed entity. The KORE dataset consists of 420 entity pairs and their relative semantic relatedness scores. This dataset mainly includes the popular entity pairs because getting judgement about the relatedness between less popular entities may require domain expertise.

Experiment

To determine the effect of taking only the annotated entities for generating a DiSER vector in high dimensional concept space, we performed experiments with a similar model i.e. ESA, which calculates the relatedness between natural language texts by using Wikipedia concepts. The main difference between DiSER and ESA is that DiSER builds the vector by taking only the annotated entities while ESA takes the full article content. We implemented ESA as it is described in (Gabrilovich 2006). To generate the DiSER vector, we perform a search for the given entity in our inverted index of entities, and retrieve only the top 1,000 Wikipedia concepts. We chose the top 1,000 articles, as most of the entities in the KORE dataset have less than 1,000 articles in which they occur. The tf-idf score of an entity is used to select the most relevant articles. We apply the same process in building the ESA vector by performing the search in the inverted index of article contents.

Existing methods of calculating entity relatedness utilize context associated with the given entity. We therefore conducted three experiments by utilizing the entity's context. In order to obtain the entity's context, an entity linker is required to extract the entities from an external resource. However, all of the entities in the KORE dataset have a Wikipedia page, therefore, we used the Wikipedia page to retrieve the context. Manually annotated entities are taken as context, which eliminates errors produced by the entity

[6] http://www.alchemyapi.com/

[7] http://www.zemanta.com/blog/demo/

[8] https://lucene.apache.org/

linking step. For instance, we define the context for "Apple Inc." as *iPad, iTunes, Steve Jobs, Apple Store, OS X etc.* We created the context vector by using three different methods: Vector Space Model (VSM), ESA and DiSER. To quantify the entity relatedness, a cosine score between the generated vectors is calculated.

To generate the context vector using VSM, we consider every appearing concept in the context as a dimension. This approach is similar to the best performing state of the art method KPCS (Hoffart et al. 2012). However, KPCS defines the magnitude of the dimensions by using Mutual Information (MI-weight) that captures the importance of the concepts for a given entity. In the second experiment, we built the ESA vector of retrieved context by considering a bag of words approach. As ESA does not distinguish between words and entities, we performed a search with each individual word. We identified that some of the retrieved articles by ESA for a given entity "Apple Inc." were completely irrelevant, which is due to words like "jobs" or "store" appearing in the context of "Apple Inc.". Similarly, we built the DiSER vector by taking the retrieved context as a bag of concepts.

We computed semantic relatedness scores for all entity pairs provided by the KORE gold standard. These scores were obtained from all five experiments: ESA, DiSER, Context-VSM, Context-ESA, and Context-DiSER. Since the gold standard dataset consists of human judgement about the ranking of 20 entities for each seed entity, we can only quantify the entity relatedness by obtaining similar judgement about the rankings from our experiments. Therefore, we calculated Spearman Rank correlation between the gold standard dataset and the results obtained from different experiments.

Entity Relatedness Measures	Spearman Rank Correlation with human
ESA	0.661
DiSER	**0.781**
Context-VSM	0.637
Context-ESA	0.684
Context-DiSER	**0.769**
WLM	0.610
KPCS	0.698
KORE	0.673

Table 2: Spearman rank correlation of relatedness measures with gold standard

Results and Discussion

Experimental results are shown in Table 2. Context-VSM is the measure that computes VSM-based relatedness score between entities context. Similarly, Context-ESA and Context-DiSER are the approaches that calculate an ESA score and DiSER score between entity context vectors. WLM is the Wikipedia Link-based approach by Witten and Milne (2008). KPCS and KORE are the approaches proposed in (Hoffart et al. 2012), where KPCS is the cosine similarity on MI-weighted keyphrases and KORE represents the keyphrase overlap relatedness. These keyphrases can be the entities appearing in context. Therefore, KPCS is a similar approach to Context-VSM. Besides, KPCS assigns MI-weights to capture the generality and specificity of entities in the context.

Many entities in the gold standard dataset are defined by ambiguous surface forms such as "NeXT" and "Nice", or they have ambiguous text segments in their surface forms like "Jobs" in "Steve Jobs" and "Guitar" in the "Guitar Hero" video game. Therefore, the effect of building a distributional vector by only considering annotated entities can be observed with the remarkable difference between the results obtained by ESA and DiSER. These scores illustrate that ESA fails in generating the appropriate distributional vector for ambiguous terms. Context-VSM does not capture the semantics of entities appearing in the context and calculates the relatedness scores by taking the overlap between these entities. Context-ESA creates the semantic vector of context, therefore, it improves the accuracy over Context-VSM. KPCS and KORE achieved significantly higher accuracy in comparison to Context-VSM, which indicates that generality and specificity of entities in the context are very influential features for entity relatedness measures.

Context-DiSER improved the accuracy of the entity relatedness measure by 10-15% over state of the art methods KPCS and KORE. KPCS captures the semantics of an entity by considering entities or keyphrases in the context. However, the entities in the context do not cover enough background knowledge to define the given entity. Therefore, it leads to the problem of topic mismatch in vector comparison. For instance, "Apple llc[9]" does not occur in the "Apple Inc." Wikipedia article. However, the DiSER vector of the context of "Apple Inc." retrieves "Apple llc" as a dimension. On the other hand, KPCS or KORE cannot capture the weakly related entities whose appropriate relatedness can be quantified only by considering a greater amount of background knowledge. This can be a major reason for getting significant improvement over KPCS, as Context-DiSER is able to capture the relatedness between weakly related entities such as "Microsoft" and "Helmut Panke".

DiSER achieved the best correlation with human ranking in this dataset. However, Context-DiSER also achieved similar accuracy scores. The KORE dataset consists of popular entities, therefore DiSER can exploit the distributional knowledge from Wikipedia. However, it may fail to obtain significant background information about the long tail entities. Instead, Context-DiSER may perform better by building the distributional vector of popular entities found in the context of the long tail entity.

Evaluation in Entity Disambiguation

Entity Disambiguation can be defined as disambiguating an entity that maps an ambiguous mention to an entity defined in a knowledge base. Mentions are the potential spans of text which can be mapped to an entity. A named entity recognition tool like Stanford NER can be used to obtain the mentions. Many different methods (Cucerzan 2007; Kulkarni et al. 2009; Hoffart et al. 2011) based on collec-

[9] http://en.wikipedia.org/wiki/Apple_IIc

tive inference mapping have been proposed. These methods jointly map several entities together in a related entity space by using entity relatedness measures. For instance, there are three mentions “Desire”, “Harris” and “Joey” in a sentence *“Desire contains a duet with Harris in the song Joey.”*; these mentions would be mapped to “Desire (Bob Dylan album)”, “Emmylou Harris” and “Joey (Bob Dylan song)” as they are related to each other.

Dataset

We performed experiments to evaluate if our approach for entity relatedness can improve the accuracy of state of the art methods which use entity relatedness measures for the entity disambiguation task. Hoffart et al. (2012) showed that different entity relatedness measures obtained significant difference in accuracy for short sentences in comparison to long news documents. Therefore, we used the KORE50 (Hoffart et al. 2012) dataset which consists of 50 short sentences with highly ambiguous mentions. There are only 14 words and nearly 3 mentions per sentence. Every mention has around 631 candidates on average to disambiguate. Sentences contain non-popular entities which have very few incoming links. As we evaluated our approach for entity disambiguation, we assume that all the mentions are given.

Experiment

We applied different entity relatedness measures to calculate relatedness scores of all the candidates of a mention to the candidates of other mentions. For instance, to find entities for the mentions “Desire”, “Harris” and “Joey”, we calculate relatedness scores of all the candidates of “Desire” to all the candidates of “Joey” and “Harris”. We use the “AIDA-Means” (Hoffart et al. 2011) dictionary to find out the candidates for a mention; it contains 35 candidates for “Desire”, 267 candidates for “Joey” and 1043 candidates for “Harris”. We obtain the confidence scores for each set of candidates by multiplying the relatedness scores of individual candidate pairs. As an example computation, we multiply the relatedness scores of “Desire (Bob Dylan album)” and “Harris (Emmylou Harris)”, “Desire (Bob Dylan album)” and “Joey (Bob Dylan song)”, and “Harris (Emmylou Harris)” and “Joey (Bob Dylan song)”, to get the final confidence score of the candidate set {“Desire (Bob Dylan album)”, “Harris (Emmylou Harris)”, “Joey (Bob Dylan song)”}. In order to get the best set of entities, we need to calculate 9.7 millions (35x267x1043) different scores. We evaluated three different approaches of entity relatedness to jointly map the entities: Joint-Context-VSM, Joint-ESA and Joint-DiSER, which use Context-VSM, ESA and DiSER respectively for entity relatedness. Since, these approaches need to calculate relatedness scores between all the candidates, it may take a long time to compute the confidence scores for very ambiguous mentions. For instance, we need to calculate more than 3 billion different confidence scores for the mentions “Steve”, “Bill”, “Sergey” and “Larry” in a given sentence *“Steve, Bill, Sergey, and Larry have drawn a great deal of admiration these days for their pioneering successes that changed the world we live in”*. Additionally, these approaches only make use of the mentions and their candidates, thus not utilizing the text around the mentions. Therefore, we performed a candidate selection by calculating mention-entity relatedness. We rank all the candidates of a given mention by calculating the ESA score between the candidate and the text around that mention in the given sentence. We select top 10 candidates for each mention and perform joint mapping by using Joint-DiSER, which we refer to as Joint-DiSER-TopN.

Results and Discussion

Table 3 shows the results of our approach. Similar to (Hoffart et al. 2012), we calculate micro-averaged and macro-averaged precisions. Micro-averaged is aggregated over all mentions in the dataset and macro-averaged is aggregated over all sentences in the dataset. Results show that different entity relatedness measures effect the accuracy of the entity disambiguation task. Joint-Context-VSM, Joint-ESA and Joint-DiSER achieved an accuracy in the same order as that of the entity ranking task, which demonstrates a consistency in results. AIDA (Hoffart et al. 2011) combines three different features: popularity, mention-entity relatedness, and entity-entity relatedness. AIDA-WLM (Hoffart et al. 2011), AIDA-KORE and AIDA-KPCS (Hoffart et al. 2012) use the WLM (Witten and Milne 2008), KORE, KPCS respectively, to perform entity relatedness. Although, AIDA-WLM and AIDA-KPCS use entity relatedness in a combination of other features, Joint-DiSER outperforms them. This shows that DiSER stands as an important feature in performing entity disambiguation. Joint-DiSER-TopN achieved the best precision and improved around 10% over state of the art methods, which shows that performing disambiguation in filtered candidates affects the performance significantly. As Joint-DiSER-TopN performs disambiguation only for selected candidates, it extremely reduces the performance time. For instance, it computes only 10K confidence scores in comparison to 3.1 billions scores for the mentions “Steve”, “Bill”, “Sergey”, and “Larry”.

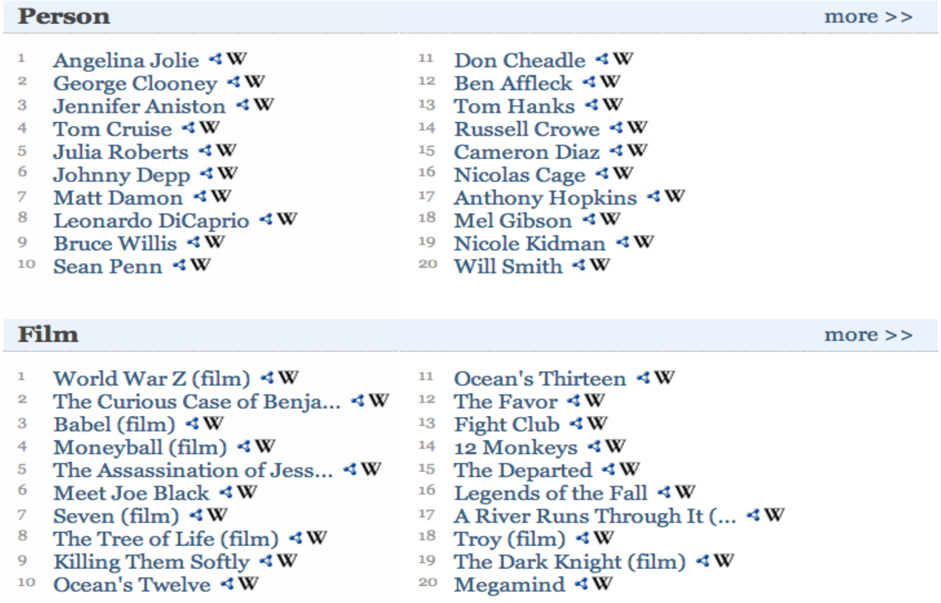
Entity Relatedness Mesures	**Micro Avg. Precision**	**Macro Avg. Precision**
Joint-Context-VSM	35.42%	34.66%
Joint-ESA	52.41%	51.74%
Joint-DiSER	58.33%	57.45%
Joint-DiSER-TopN	**71.83%**	**70.10%**
AIDA-WLM	57.64%	56.00%
AIDA-KORE	64.58%	62.60%
AIDA-KPCS	55.64%	54.70%

Table 3: Entity Disambiguation accuracy on KORE50 dataset

Entity Relatedness Graph

To demonstrate our approach we implemented the EnRG (Entity Relatedness Graph)[10] web application. EnRG is constructed by calculating the DiSER scores between 16.83 trillions of entity-pairs (4.1 millions x 4.1 millions). DiSER

[10] http://server1.nlp.insight-centre.org:8080/enrg/



	Person
1	Angelina Jolie
2	George Clooney
3	Jennifer Aniston
4	Tom Cruise
5	Julia Roberts
6	Johnny Depp
7	Matt Damon
8	Leonardo DiCaprio
9	Bruce Willis
10	Sean Penn
11	Don Cheadle
12	Ben Affleck
13	Tom Hanks
14	Russell Crowe
15	Cameron Diaz
16	Nicolas Cage
17	Anthony Hopkins
18	Mel Gibson
19	Nicole Kidman
20	Will Smith

more >>

	Film
1	World War Z (film)
2	The Curious Case of Benja...
3	Babel (film)
4	Moneyball (film)
5	The Assassination of Jess...
6	Meet Joe Black
7	Seven (film)
8	The Tree of Life (film)
9	Killing Them Softly
10	Ocean's Twelve
11	Ocean's Thirteen
12	The Favor
13	Fight Club
14	12 Monkeys
15	The Departed
16	Legends of the Fall
17	A River Runs Through It (...
18	Troy (film)
19	The Dark Knight (film)
20	Megamind

more >>

Figure 1: Ranked list of people and films related to Brad Pitt

builds the vector by taking top 1000 articles from 4.1 millions of articles. It can be seen as a sparse square matrix of order 4.1 million, and all the scores in every row except the top 1000 scores, converge to zero. We have to calculate the cosine scores between all the rows. In order to calculate the 16.83 trillion scores, a very fast and efficient computing is required. Even if the system is able to process more than half a millions entity-pairs per second, the complete process will take more than a year. Therefore, we applied a pruning technique which only calculates the DiSER score if it would be a non-zero value. We collect all the possible related entities with non-zero scores for a given entity. Since DiSER takes only the top 1,000 articles to build the vector, the entities not appearing in the content of the top 1,000 articles for a given entity would produce a zero relatedness score with that entity. For instance, if DiSER takes only the top 2 articles to calculate the relatedness score, and we want to retrieve all the entities having a non-zero relatedness score with "Apple Inc.", we would obtain all the entities such as "Steve Jobs", "iPad" and "OS X" as they appear in the content of top 2 articles of "Apple Inc." and would not retrieve entities like "Samsung" and "Motorola" as they do not appear in top 2 articles. We obtained around 10K related entities for every individual entity. Therefore, we calculate DiSER scores for only 4.1 billions of entity-pairs, and this reduces the comparisons by 99.8%. Our system takes around 48 hours to build the EnRG graph with 25K comparisons per second.

Similar to the major search engines, the EnRG graph provides a ranked list of related entities for a given entity. It provides different aspects of related entities by categorizing them into different classes using their DBpedia type. Therefore, a user can obtain a ranked list of different types of related entities such as "Person", "Film", "Company" and others. Figure 1 shows a snapshot of our EnRG interface, which illustrates a ranked list of related people and films to Brad Pitt.

Conclusion and Future Work

We presented DiSER to compute semantic relatedness between entities. We used Wikipedia as it consists of world knowledge about millions of entities. DiSER builds distributional vectors by taking only the manually annotated entities appearing in Wikipedia articles. Therefore, it can build unambiguous distributional vectors for ambiguous surface forms of the given entities. In our experiments, DiSER outperforms state of the art methods and achieves a significant improvement in entity ranking and disambiguation tasks over other methods. We also proposed an alternative Context-DiSER approach to generate the DiSER vectors for long tail and non-popular entities, which do not have a Wikipedia page. We also discussed a web application EnRG graph, which can be used to retrieve a ranked list of related entities with their types for a given entity in real time. Future work will be in applying DiSER and EnRG in various tasks like information retrieval (semantic search), text mining and knowledge base population.

Acknowledgments

This work has been funded in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (INSIGHT) and by the EU FP7 program in the context of the project LIDER (610782).

References

Aggarwal, N., and Buitelaar, P. 2012. Query expansion using wikipedia and dbpedia. In Forner, P.; Karlgren, J.; and Womser-Hacker, C., eds., *CLEF*.

Aggarwal, N.; Asooja, K.; and Buitelaar, P. 2014. Exploring esa to improve word relatedness. *Lexical and Computational Semantics (* SEM 2014)* 51.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer. 722–735.

Banerjee, S., and Pedersen, T. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational linguistics and intelligent text processing*. Springer. 136–145.

Blanco, R.; Cambazoglu, B. B.; Mika, P.; and Torzec, N. 2013. Entity recommendations in web search. In *International Semantic Web Conference (2)*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD*, 1247–1250. ACM.

Cilibrasi, R. L., and Vitanyi, P. M. 2007. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on* 19(3):370–383.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, 708–716.

Demartini, G.; Missen, M. M. S.; Blanco, R.; and Zaragoza, H. 2010. Taer: time-aware entity retrieval-exploiting the past to find relevant entities in news articles. In *Proceedings of the 19th ACM CIKM*.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, 1606–1611.

Gabrilovich, E. 2006. *Feature generation for textual information retrieval using world knowledge*. Ph.D. Dissertation, Technion - Israel Institute of Technology, Haifa, Israel.

Gale, W. A.; Church, K. W.; and Yarowsky, D. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics.

Harris, Z. 1954. Distributional structure. In *Word 10 (23)*, 146–162.

Hassan, S., and Mihalcea, R. 2011. Semantic relatedness using salient semantic analysis. In *AAAI*.

Hirst, G., and St-Onge, D. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database* 305:305–332.

Hoffart, J.; Yosef, M. A.; Bordino, I.; Fürstenau, H.; Pinkal, M.; Spaniol, M.; Taneva, B.; Thater, S.; and Weikum, G. 2011. Robust disambiguation of named entities in text. In *Proceedings of the EMNLP*, 782–792.

Hoffart, J.; Seufert, S.; Nguyen, D. B.; Theobald, M.; and Weikum, G. 2012. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM CIKM*, 545–554. ACM.

Ji, H.; Grishman, R.; Dang, H. T.; Griffitt, K.; and Ellis, J. 2010. Overview of the tac 2010 knowledge base population track. In *TAC 2010*.

Kulkarni, S.; Singh, A.; Ramakrishnan, G.; and Chakrabarti, S. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD*.

Landauer, T. K.; Foltz, P. W.; and Laham, D. 1998. An introduction to latent semantic analysis. *Discourse Processes* 25:259–284.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Pantel, P., and Fuxman, A. 2011. Jigs and lures: Associating web queries with structured entities. In *ACL*, 83–92.

Pirró, G., and Euzenat, J. 2010. A feature and information theoretic framework for semantic similarity and relatedness. In *The Semantic Web–ISWC 2010*. Springer. 615–630.

Polajnar, T.; Aggarwal, N.; Asooja, K.; and Buitelaar, P. 2013. Improving esa with document similarity. In *Advances in Information Retrieval*. Springer. 582–593.

Ponzetto, S. P., and Strube, M. 2007. Knowledge derived from wikipedia for computing semantic relatedness. *J. Artif. Intell. Res.(JAIR)* 30:181–212.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Strube, M., and Ponzetto, S. P. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, 1419–1424.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th WWW*, 697–706. ACM.

Witten, I., and Milne, D. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, 25–30.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 133–138. Association for Computational Linguistics.

Yu, X.; Ma, H.; Hsu, B.-J. P.; and Han, J. 2014. On building entity recommender systems using user click log and freebase knowledge. In *WSDM '14*, 263–272.