What is semantically important to “Donald Trump”?

Wan, Jizheng; Barnden, John; Hu, Bo; Hancox, Peter

What is Semantically Important to "Donald Trump"?

Jizheng Wan[1,2], John Barnden[1], Bo Hu[3] and Peter Hancox[1]

[1] University of Birmingham, UK
[2] Coventry University, UK
[3] Barclays, UK
Jizhneg.Wan@coventry.ac.uk
J.A.Barnden@cs.bham.ac.uk
bo.hu@barclays.co.uk
P.J.Hancox@cs.bham.ac.uk

Abstract. In the recent years, there is a growing interest in combining explicitly defined formal semantics (in the forms of ontologies) with distributional semantics "learnt" from a vast amount of data. In this paper, we try to bridge the best of the two worlds by introducing a new metrics called the "Semantic Impact" together with a novel method to derive a numerical measurement that can summarise how strong an ontological entity/concept impinges on the domain of discourse. More specifically, by taking into consideration the semantic representation of a concept that appears in documents and its correlation with other concepts in the same document corpus, we measure the importance of a concept with respect to the knowledge domain at a semantic level. Here, the "semantic" importance of an ontology concept is two-fold. Firstly, the concept needs to be informative. Secondly, it should be well connected (strong correlation) with other concepts in the same domain. We evaluated the proposed method with 200 BBC News articles about Donald Trump (between February 2017 and September 2017). The preliminary result is promising: we demonstrated that semantic impact can be learnt: the top 3 most important concepts are Event, Date and Organisation and the least essential concepts are Substance, Duration and EventEducation. The crux of our future work is to extend the evaluation with larger datasets and more diverse domains.

Keywords: Semantic Impact, Ontology Learning, XYZ Model, Word2Vec.

1 Introduction

As a key enabling technology of Semantic Web, the concept of ontology has been widely used in, not only research labs but also large-scale IT projects. It is "a formal language designed to represent a particular domain of knowledge" [1], and as with other knowledge-based studies in computer science research, people have always dreamed of developing a self-learning mechanism to automate the generation of such formal representation.

Since Maedche and Staab coined the term "Ontology Learning" (OL) [2], people have experimented various learning approaches. Roughly, these approaches have been

grouped into four categories: statistical approach, linguistic approach, logical approach and hybrid approach [3]. However, one of the challenges amount all these approaches is that at some point of the learning process the system needs to make a decision on whether or not a particular concept should be included in the domain ontology. It is our contention that a method to measure the importance (or relevance) of a concept to the domain knowledge is essential to make such a decision across all OL methodologies.

Using "Harry Potter" as an example. Horrocks [4] demonstrated how to use RDF and OWL to describe the text below, which makes it possible for the software agent to discover that there is a `hasPet` relation between `HarryPotter` and `Hedwig`. Additional properties have been defined at a later stage, such that `HarryPotter` is a (`rdf:type`) `Wizard` and a `Student`, and that `Hedwig` is a `SnowyOwl`.

"Harry Potter has a pet called Hedwig."

Assuming that we need to build an ontology containing key concepts in the Harry Potter story. An immediate question is what concepts could be considered as key, in other words, what makes Harry Potter "Harry Potter"? Three concepts (or ontology classes) have been identified in the above example: `Wizard`, `Student` and `SnowyOwl`. Since the whole story is about how a young wizard studies magic at the Hogwarts and fights against an evil senior wizard who graduated from the same school, it is easy to understand that `Wizard` and `Student` are more "important" than the `SnowyOwl`, because without them, Harry Potter would no longer be the "Harry Potter" that we are familiar with. On the other hand, the entire story is still coherent if he has a different pet or has no pet at all. Therefore, `Wizard` and `Student` concepts have a bigger influence than the `SnowyOwl` on the domain knowledge. In this paper we use the term "*Semantic Impact*" to describe such influence.

In traditional NLP or IR study, there are various ways to measure how important (or relevant) a word is with respect to a document in the corpus. However, the importance or relevance of a word to a document at the syntax level is not quite the same as the importance or relevance of a concept to the domain knowledge at the semantic level. Using TF-IDF as an example, even if people can solve the problem that in fact the `Wizard` concept contains multiple words (e.g. Harry Potter, Lord Voldemort etc.), it is still difficult to reach a high *tf-idf* weight to compute its relevace to the corpus. Simply because it is almost guaranteed that this concept will exist in every chapter/document about Harry Potter and therefore will have a low, if not 0, *idf* value which suggests that it is not very informative at all. Previous research also suggested that in some cases, *idf* does not provide any improvement and therefore the *tf* (or a similar) scheme itself is sufficient [5]. In which case, the more a term appears in the corpus, the more relevant it is. However, it is not necessarily true at the semantic level. As demonstrated in this paper, in the news article domain, the concept of `Date` has a low frequency compare with other concepts such as `Person` and `Place`, but it can generate a more significant semantic impact comparing with the other two.

Therefore, it is unreliable to purely use the frequency or statistics-based approach to decide the "relevance" or "importance" at the semantic level. One common way to handle this issue in the OL study is by relying on some pre-defined knowledge to determine what should and should not be included in an ontology. By so doing, the system

will lose the ability to learn new concepts and in which case it is more likely to be an ontology populator rather than a learning approach.

As part of the XYZ Model research[6], this paper will introduce a new idea called the "*Semantic Impact*" to measure how valuable a concept to the domain knowledge at the semantic level. There is a mathematical definition at the end of Section 2, its textual definition is given as:

> *Semantic Impact (SI) represents how informative a concept is in the corpus and moreover the strength of its correlation with the other concepts in the domain.*

In order to accurately measure the semantic impact, a novel approach will be discussed in this paper. For demonstration purpose, we have manually collected a set of news articles, between February 2017 and September 2017, about Donald Trump and split into two corpora: Source Corpus and Target Corpus. Then use this approach to generate some interesting results about how semantically important each concept in the "Trump" domain is.

2 Research Methodology

In traditional computational linguistics study, the idea of the Distributional Semantic Models (DSM) is that the meaning of words can (at least to a certain extent) be inferred from their usage and therefore the semantics can be encapsulated in high-dimensional vectors based on the nearby co-occurrence of words [7]. There are various tools/frameworks, e.g. Word2Vec [8, 9], that have been developed to vectorise the words in the corpus so as to generate the semantic representation.

By adopting and expanding the DSM theory, this research is based on two assumptions: **a)** high-dimensional vector can also be used to infer the semantic representation of a concept, which extensionally is a set of words that belong to the same semantic group, and **b)** with sufficient data, for any concept in a domain, the distribution of its semantic representation is consistent.

Therefore, the underlying philosophy of this research is to cross-compare the semantic representation information between two corpora about the same domain. So, the system will be able to identify the patterns of the distribution for the domain concepts, then train a set of neural networks to distinguish the high informative concepts from other low informative concepts.

Moreover, it is possible to use the representation of a specific concept to measure the impact or influence that a particular word (or a list of words) could bring to it. By doing so for all the domain concepts on all the vocabularies in the corpora, the system will then be able to measure the correlation between each concept-pairs.

Let I_a be the *informative coefficient* for the concept a, C_a be the *correlation coefficient* it has with the other concepts and λ be a constant that adjusts the weight of the correlation, then its semantic impact SI_a can be calculated as follows:

$$SI_a = I_a + \lambda C_a \tag{1}$$

The value of λ is normally set empirically and depends on document corpus. For example, if a domain only contains a small number of concepts, then it is highly likely that all these concepts have a strong correlation with each other and thus the informative coefficient plays a more critical role in deciding the semantic impact. A smaller value, therefore, could be assigned to λ (e.g. 0.5) to reduce the overall contribution of C_a

2.1 System architecture

The overall process is shown in **Fig. 1**. The first step is to use an existing tool/method to extract the basic concepts and relations from the source and target corpus and convert into the associated Document based Ontology (DbO) set [6]. Then step 2 uses a normalisation and vectorisation process to generate the semantic distribution vector for all the concepts identified in the previous step. Step 3 is designed to calculate the informative coefficient (I) and then use a Maximal Information Coefficient (MIC) [10] based approach, in Step 4, to analyse the correlations between each class/concept pair and generate the correlations coefficient (C). Finally, in Step 5, to **Equ. 1** to calculate the semantic impact value. The following section will discuss these steps in detail.

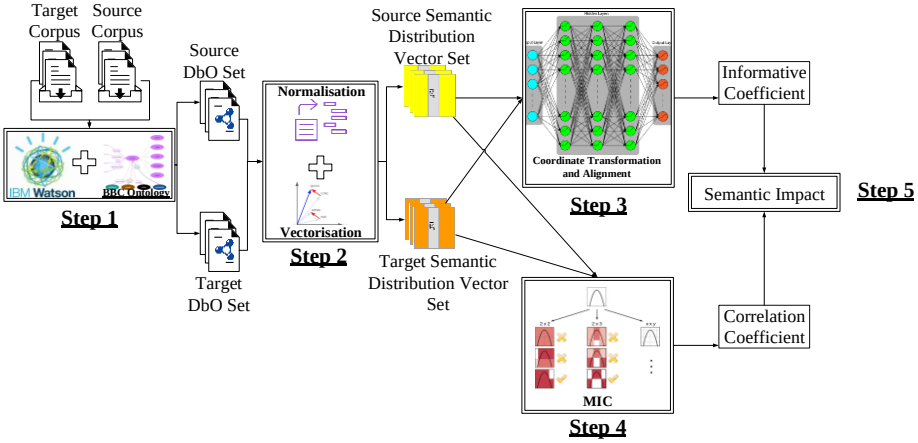

Fig. 1. Process Overview

2.2 DbO Construction (Step 1)

As introduced in [6], DbO is an ontology that operates on the document level without concern for the wider context. Essentially, 200 news articles about Donald Trump was manually collected from the BBC News website and split into two corpora: Source Corpus and Target Corpus. Subsequently, the IBM Natural Language Understanding (NLU) [11] service with the default News annotation model was selected to analyse these documents and extract various semantic information (concepts and relations) from them. For example, the below "Relations" information is one of the 438 relations that have been identified from one article [12] by this process.

Relation


```
{
  "type": "agentOf",
  "firstEntityType": "Person",
  "secondEntityType": "EventCommunication",
  "secondEntity": "said",
  "firstEntity": "Sean Spicer",
  "sentence": "Before the list was published, press secretary Sean
Spicer said there were \"several instances\" of attacks that had not gained
sufficient media coverage (without specifying which fell into that
category).",
  "score": "0.99692"
},
```


After this semantic information extraction process, there is a class/property mapping process (DbO/O Mapping) to manually map the `Entity Types` and `Relation Types` to the `Class` and `Property` in the BBC Core Concepts Ontology [13] which is the initial ontology we use to generate the benchmark for further analysis. Then, the system will generate a DbO set based on these relations and the mapping information. For instance, if an `Entity Type` has a linked Ontology Class (in the BBC Core Concepts Ontology), then the system will automatically inherit the properties and relations (that also exist in this DbO) that are defined in the BBC Core Concepts Ontology and use this inherited information to construct the DbO. If mapping does not exists, the system produces a new empty `Class` and add it into the DbO.

In the following example, `Entity Type` will be considered as the ontology class; `Relation Type` will be treated as the ontology property, and `Relations` will be converted into the ontology `Individuals`:

Ontology Class

```
DbO:Weapon  a              owl:Class ;
         rdfs:isDefinedBy
DbO:0e0e6bf58a95f44aee0f937e33a2532b ;
         rdfs:label        "Weapon"@en .
```

Ontology Property

```
DbO:occupation  a           owl:ObjectProperty ;
        rdfs:domain         DbO:Person ;
        rdfs:isDefinedBy
DbO:0e0e6bf58a95f44aee0f937e33a2532b ;
        rdfs:label          "occupation"@en ;
        rdfs:subPropertyOf  DbO:notablyAssociatedWith .
```

Ontology Individual

```
DbO:15d64395d922fa5ff25ba4b01b0f9615_0.726746
        a                           DbO:NaturalEvent ,
DbO:Vehicle ;
        DbO:eventTheme              "affectedBy" ;
        Property:FirstEntity        "vehicle" ;
        Property:FirstEntityType    "Vehicle" ;
        Property:Score              "0.726746" ;
        Property:SecondEntity       "crashed" ;
        Property:SecondEntityType   "NaturalEvent" ;
        Property:Sentence           "Kuwait City, Kuwait, October
2016 What happened: An Egyptian man was detained after a bin
lorry reportedly loaded with explosives crashed into a vehicle
carrying five US soldiers." .
```

By going through all the documents in the Source and Target Corpus, the system generates 200 DbOs that are grouped into two sets: Source DbO Set and Target DbO Set. Compared with the original information extracted from the NLU, the DbO set is more convenient for us to analyse the ontological relations between each of the individuals.

2.3 Semantic Distribution Calculation (Step 2)

Vectorisation is done by the Word2Vec model in the DeepLearning4J framework [14] with the following configuration: `MinWordFrequency` = 1, `LayerSize` = 100 and `WindowSize` = 5.

The semantic distribution is, in fact, a vector obtained from the vectorisation process. It is easy to get the semantic distribution for any single word in the corpus, but since a concept will contain multiple words, the challenge here is how to generate a single vector to represent the collection of individual word vectors that preserve the semantic meaning of the concept in a high-dimension space.

This is achieved by a normalisation process. The basic idea is to replace all the relevant words/entities about a specific concept from the corpus with a unique string and re-run the vectorisation process to generate a new Word2Vec model for this specific concept. Then the vector of this unique string could be considered as a projection of all the vectors of the replaced words on this newly created Word2Vec model, and considered to be tantamount to semantic distribution vector for the original concept/class. By repeating this process, we could generate a separate Word2Vec model for all the concepts in both Source and Target Corpus respectively. We denote the new Word2Vec models created via this normalisation process `W2V_<ConceptName>` and the original Word2Vec model generated from the corpus `Master W2V`. Meanwhile, we use the `Master W2V` as the baseline model for aligning `W2V_<ConceptName>` models discussed in the next section.

There are two reasons to generate separate models instead of replacing all the relevant words from the corpus with all the unique strings in one go. Firstly, by the nature of how Word2Vec (or any word embedding method) works, replacing too many words may significantly change the grouping structure, and therefore the new model will not

be able to represent the same semantic distribution as the old model does. Hence, it is essential to minimise the amount of words that need to be replaced in each model in order to maximise the consistency of the semantic representation.

Secondly, within the different context, the same word could be identified as different concepts. For example, the word "Trump" can be both Person and Place (the Trump building), and we cannot replace the same word twice with two different unique strings in one model.

2.4 Coordinate Transformation (CT) Process (Step 3 – Part 1)

By the end of the last process, the system generates two sets of the semantic distribution vector as well as the associated Word2Vec models. Essentially, the system will use these vectors to calculate the informative coefficient information. However, before giving further details, there is a more general issue that needs to be discussed here: how to compare vectors that occur in two different Word2Vec models.

Word2Vec is one of the most popular methods to vectorise the words in the corpus and generate the semantic representations [8, 9] and Cosine Similarity (CS) is one of the primary methods used to compare two words/vectors inside one Word2Vec model. However, most of the vectors used by this research are in fact from different Word2Vec models and effectively projected upon to different coordinate systems whereof CS values cannot be calculated directly. It is essential to perform coordinate transformation to align different Word2Vec models. This alignment can be anchored on common words appearing in both models.

For example, if both Word2Vec models XYZ and X'Y'Z' have words "Trump" and "President", let $\vec{V}_1^T$ and $\vec{V}_1^P$ be the vector of the word "Trump" and "President" in the first model respectively and $\vec{V}_2^T$ and $\vec{V}_2^P$ be the corresponding vectors in the second model, the goal is to make $\vec{V}_1^T$ and $\vec{V}_2^T$ as close to each other as possible (same applies to $\vec{V}_1^P$ and $\vec{V}_2^P$). This is formally defined as follows:

$$\text{Argmax}\left(\frac{\vec{V}_1^T \cdot \vec{V}_2^T}{\|\vec{V}_1^T\|\|\vec{V}_2^T\|} + \frac{\vec{V}_1^P \cdot \vec{V}_2^P}{\|\vec{V}_1^P\|\|\vec{V}_2^P\|}\right) \tag{2}$$

We simplify the solution to the above formula to a classic supervised learning problem with neural network. Let XYZ be the master or target Word2Vec model (to be aligned against) and X'Y'Z' be the source model (align from). Also let $\vec{V}_2^T$ & $\vec{V}_2^P$ as the input, and $\vec{V}_1^T$ & $\vec{V}_1^P$ the labels of the associated input.

The neural network implementation consists a fully-connected feedforward neural network with 3 hidden layers as illustrated in **Fig. 2**. It takes a 100×1 vector (LayerSize of the W2V) as the input and outputs a 100×1 vector. We use TANH on the Output Layer forcing the output values to scale down to between [-1,1]. Each of the Hidden Layers contains 2000 nodes and uses ReLU as the activation function. The other configurations include:

- Using XAVIER for the weight initialisation [15].
- Using ADAM as the method for the stochastic optimisation [16].

- Set to BatchSize 100.
- Set to Number of Epochs 350.

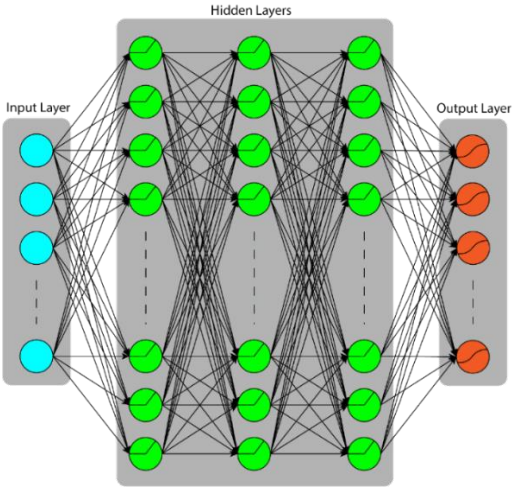

Fig. 2. Neural Network Structure for the CT Process

2.5 Aligned Cosine Similarity and Informative Coefficient (Step 3 – Part 2)

As discussed before, there is a difference between the frequency-based relevance at the literal level and the informative at the semantic level. This difference is caused by the fact that the former does not take into consideration the position of a word in the sentence and its context while the second one does. Since the Semantic Distribution Vector (SDV) for a specific concept is created by the normalisation process, which is essentially a projection of all the related word vectors, their informative complexity will be inherited in the SDV which is included in the `W2V_<ConceptName>` model.

This section will focus only on the process we use to generate the informative coefficient (I): the reason for this will be explained in the next section.

For a specific Concept/Class a, let CS'_a be the Aligned Cosine Similarity, and $\overline{Conf_a}$ be the average confidence score. Then:

$$I_a = CS'_a \times \overline{Conf_a} \tag{3}$$

Fig. 3 shows how to calculate the CS'_a. Using the Event Class as an example, `Source W2V_Event` is the Word2Vec model generated by the normalisation process from the source domain and $\vec{V}_S$ is the `Semantic Distribution Vector` for the `Event` concept/class in this model. So, by using the Coordinate Transformation Process discussed before, Step 3.1 aligns the `Source W2V_Event` model with the `Source Master W2V` model to create the `Aligned Source W2V_Event` model and the aligned distribution vector $\vec{V}'_S$. Step 3.2 applies a similar process to align the `Target W2V_Event` model with the `Target Master W2V` model, to create the `Aligned Target W2V_Event` model and the aligned distribution vector $\vec{V}'_T$. In

Step 3.3, the system aligns the `Target Master W2V` model with the `Source Master W2V` model and create an `Interim W2V` model. Then Step 3.4 will align the `Aligned TargetW2V_Event` model, which was created in Step 3.2, with this `Interim W2V` model to create an `Aligned Interim W2V` model which contains a new aligned distribution vector $\vec{V}'_{TS}$. Finally, the Cosine Similarity between $\vec{V}'_S$ & $\vec{V}'_T$ (CS_{Event}) and $\vec{V}'_S$ & $\vec{V}'_{TS}$ (CS'_{Event}) is calculated as:

$$CS_{Event}(\vec{V}'_S, \vec{V}'_T) = \frac{\vec{V}'_S \cdot \vec{V}'_T}{\|\vec{V}'_S\| \|\vec{V}'_T\|} \qquad CS'_{Event}(\vec{V}'_S, \vec{V}'_{TS}) = \frac{\vec{V}'_S \cdot \vec{V}'_{TS}}{\|\vec{V}'_S\| \|\vec{V}'_{TS}\|} \tag{4}$$

By enumerating all the `Ontology Individuals` which contain at least one Event class in the DbO set, it is easy to get the sum of the score (`Property:Score` in the Individual) which is the relation confidence score gets from the IBM NLU process and ranging from 0 (not confident) to 1 (highly confident). Let N_{Event} be the total number of such `Ontology Individual`, then $\overline{Conf}_{Event}$ will be:

$$\overline{Conf}_{Event} = \frac{\sum_{i=0}^{n} Score}{N_{Event}} \tag{5}$$

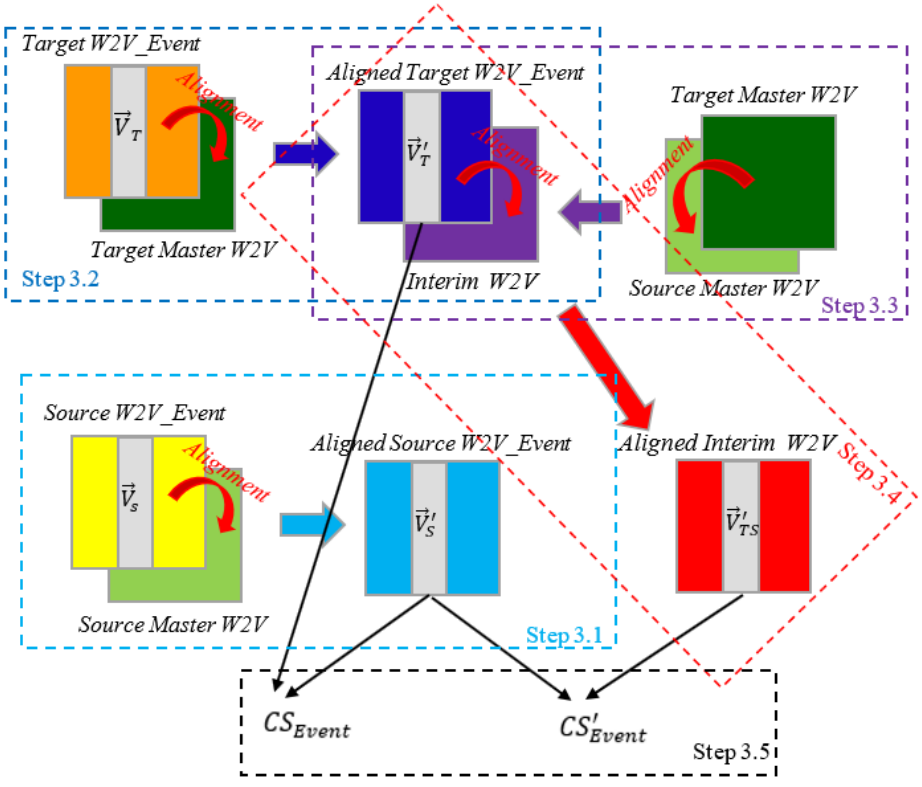

Fig. 3. Cosine Similarity and Aligned Cosine Similarity

2.6 MIC-based Correlation Analysis (Step 4) and the Final Result (Step 5)

Maximal Information Coefficient (MIC) was introduced by David Reshef to measure the strength of the linear or non-linear association between two variables [10]. It can be used to not only identify essential relationships in the dataset but also to characterise them.

Consider a Domain Ontology as a function which could be used to represent knowledge within a domain. Then the Ontology Classes will be the variables of this function. Moreover, individual words that exist in the corpus are the essential components and "material" that build the domain knowledge. Therefore, each word will have an influence on the knowledge that the Ontology represents. Hence, the individual word will have an indirect impact on the Ontology manifested through the Classes that the individuals belong to. As a result, if we could measure the impact a word could exercise on the various Classes, we could then understand the relations between these Classes. In other words, considering each word in the corpus as an independent sample, the Classes are the variables (or properties), and the value of a specific variable/property in a specific sample is the Cosine Similarity between that word and that concept/class. In this way, the system can generate a sample table where each row corresponding to a word in the Word2Vec vocabulary list, and each column corresponding to a concept/class that has been identified in the corpus.

Using the sample table as the input, the MIC algorithm generates the result that indicates the strength of the correlation between all the class pairs. The correlation coefficient for concept/class a can then be calculated as:

$$C_a = \log\left(\sum_{i=0}^{|R_a|} MIC(a,b) \times \overline{Conf}(b_i) \times \overline{Conf}(a)\right) \tag{6}$$

where $R_a = \{\langle a,b\rangle | \exists b, \langle a,b\rangle \in R\}$.

Therefore, the completed formula for the semantic impact is (Step 5):

$$SI_a = \frac{\vec{V}_S' \cdot \vec{V}_{TS}'}{\|\vec{V}_S'\|\|\vec{V}_{TS}'\|} \times \frac{\sum_{i=0}^{n} Score(a)}{N_a} + \lambda \log\left(\sum_{i=0}^{|R_a|} MIC(a,b) \times \frac{\sum_{i=0}^{n} Score(b_i)}{N_b} \times \frac{\sum_{i=0}^{n} Score(a)}{N_a}\right) \tag{7}$$

3 Results and Discussion

We carried out a preliminary evaluation wherein 35 `Entity Types` across the Source and Target Corpus have been identified using IBM NLU service. By going through manual mapping, the 35 entity types have been converted into 29 concepts (or ontology classes) in the DbO Sets as listed in **Table 1**.

Table 1. Concepts/Ontology Classes in the DbO Sets

Award	Cardinal	Crime	Date
Duration	EntertainmentAward	Event	EventBusiness
EventCustody	EventDemonstration	EventEducation	EventElection
EventPerformance	EventPersonnel	EventViolence	Facility

GeographicFeature	HealthCondition	NaturalDisaster	Organisation
Person	Place	Product	SportingEvent
Substance	Time	TitleWork	Vehicle
Weapon			

After the Normalisation and Vectorisation process (Step 2), each of them had an associated `Semantic Distribution Vector`. By going through the Step 3 in the **Fig. 1**, the system will be able to generate their CS, CS' and the informative coefficient.

Table 2. Cosine Similarity, Informative Coefficient and Term Frequency (sorted by the CS')

Concept/Class	$\boldsymbol{CS}$	$\boldsymbol{CS'}$	$\boldsymbol{I}$	**Total TF**
Event	9.05786E-05	0.932403684	0.762811831	20.00957069
Organisation	-0.074791484	0.874482393	0.59440452	10.37735849
Place	0.155734465	0.838355482	0.566655526	13.93218485
Date	-0.146419838	0.816355526	0.66578382	2.782335247
Cardinal	-0.072149187	0.772277176	0.4999575	0.676784249
EventViolence	-0.067453243	0.65089637	0.408211891	0.929723817
EventPerformance	0.03419451	0.592700899	0.419367313	0.478534318
EventPersonnel	0.072302915	0.466879278	0.32830291	1.004922067
Person	-0.043701328	0.456888855	0.323156035	34.1878589
EventCustody	0.016087731	0.293029428	0.203772185	0.464861909
EventBusiness	0.012715162	0.27680552	0.16905362	0.006836205
NaturalDisaster	-0.061361331	0.190954998	0.085705599	0.116215477
Weapon	0.016315045	0.170783401	0.075626779	0.389663659
GeographicFeature	0.036188241	0.124441072	0.061816507	0.355482636
SportingEvent	-0.145656377	0.113705434	0.078391696	0.683620454
EntertainmentAward	-0.109663352	0.089443691	0.052318755	0.116215477
EventElection	0.069957979	0.087879911	0.052950434	1.196335794
Product	-0.09157607	0.080848917	0.047825187	0.006836205
EventDemonstration	-0.041675355	0.0531593	0.024467885	0.102543068
Facility	0.118815102	0.04608589	0.030354911	2.119223407
Duration	0.115452491	0.015852489	0.005771626	0.047853432
HealthCondition	-0.126119331	0.01493654	0.011196306	0.546896363
Award	-0.042326197	0.009894854	0.005180831	0.109379273
Vehicle	-0.105587758	-0.007998363	-0.004870428	0.403336068
TitleWork	0.08005926	-0.07365784	-0.04931166	0.102543068
Time	-0.196784243	-0.079296142	-0.053095143	0.129887886
Crime	0.011954751	-0.079479031	-0.05448308	0.334974022
Substance	-0.076414958	-0.092376187	-0.026769871	0.034181023
EventEducation	-0.009967238	-0.177876234	-0.086416084	0.020508614

There are four classes in the BBC Ontology: `Event`, `Organisation`, `Place` and `Person`. For now, we refer to them as `Ontology Class`. The rest of the classes in the above table are identified by the IBM NLU process and will be called `Candidate Class` in order to distinguish from the former. It is interesting to see from the above result that all the `Ontology Classes` have a high CS' value. In fact, a positive correlation between the CS' value and the informative coefficient is evidenced.

An intuitive explanation is as follows. It is easy to understand that for a class with a high informative coefficient value, such as an `Ontology Class`, it will have a more complex structure and relation (or contains more semantic information) compared to a class with a low informative coefficient value. As discussed already, this complexity will be inherited, during the normalisation process, in its semantic representation, and therefore its final *Semantic Distribution Vector* will be more "complex" (or contain more semantic information) than the distribution vector for a class with low IC value even if they have the same dimension size (100x1). Moreover, when we use the Coordinate Transformation Process to align the `W2V_<ConceptName>` Model with the Source (or Target) `Master W2V` Model (Step 3.1 and Step 3.2 in the **Fig. 3**), it is in fact using a neural network to predicate a vector for a word (the unique string) that never existed in the original `Master W2V` Model. As a result, the CS' value in **Table 2** is essentially the degree of alignment of the predication. With this idea in mind, the above result suggests that this predication and alignment process only works well on those classes with a high informative value, otherwise, their CS' value should all be close to 1.

In order to eliminate the possibility that the individual concept/class neural network never been trained properly, we have calculated the CS and CS' for all the overlapping vocabularies in the related two Word2Vec models, which is the training data set, and then calculated its average (as shown in **Table 3**).

If these two models are perfectly aligned with each other, then the average value after the alignment should be equal to 1. The result clearly shows that all the neural networks are "properly" trained and work extremely well on the training dataset. This suggests that the neural network trained for a class with a low informative value may be subject to overfitting due to the simplicity of the problem it is trying to solve. However, when comes to the class with high informative value, the problem complicity helps to reduce the chance of overfitting and leads to a more accurate "good" result. A positive side-effect is that the overfitting-ness could be used as a criterion to distinguish the low informative concepts/classes from the high informative concepts/classes.

Table 3. Neural Networks Evaluation Result

Neural Network	**After Alignment** (CS')	**Before Alignment** (CS)
Award	0.98031644	0.07930794
Cardinal	0.97966864	0.03505877
Crime	0.98225026	0.06569504
Date	0.98123691	0.02029948
Duration	0.97950185	0.09908933
EntertainmentAward	0.97891328	0.10126378
Event	0.98852654	0.02643711
EventBusiness	0.9786953	0.09117983
EventCustody	0.97857699	0.08236471
EventDemonstration	0.97864903	0.09305801
EventEducation	0.97960562	0.09103634
EventElection	0.97835062	0.04980883
EventPerformance	0.9785876	0.08383297
EventPersonnel	0.97807607	0.05734759

EventViolence	0.98036507	0.06654035
Facility	0.98025752	0.03982984
GeographicFeature	0.98052347	0.08840587
HealthCondition	0.97929321	0.07387248
NaturalDisaster	0.97913864	0.0762092
Organisation	0.97541052	0.0213745
Person	0.99137417	0.00545538
Place	0.98766889	0.02258751
Product	0.97633725	0.09864551
SportingEvent	0.97820687	0.08318475
Substance	0.97602775	0.08991432
Time	0.98071897	0.08842963
TitleWork	0.97798596	0.08760807
Vehicle	0.98034503	0.07701601
Weapon	0.97641792	0.07500139

Since there are 29 concepts/classes identified from the corpus, we have 406 class pairs in total and the correlation analysis process will generate a MIC strength value for each of the pairs. Due to the reason of size, **Table 4** lists only the top 10 pairs. Based on the MIC result, it is easy to calculate the correlation coefficient value for all these 29 concepts (Formula 6) in the Source Corpus (Step 4) and then will be able to get the final semantic impact value which shows in **Table 5** (Step 5). In this demonstration, we consider the informative and correlation are equally important and therefore $\lambda = 1$.

From the result, we can clearly see that `Event` (e.g. reported, announced and promise), `Data` (e.g. today, yesterday and next week) and `Organisation` (e.g. united nation, council and republican) are the most important concepts/classes in the domain, due to the high *Informative* and *Correlation Coefficient* values. On the other hand, `EventEducation` (e.g. graduating and graduated), `Duration` (e.g. 22-minute, 80-minute and more than a year) and `Substance` (e.g. steel and coal) are the least important concepts.

It is interesting to see that the concept `Date` has a relatively low TF value but with a high *Semantic Impact* value as a result of both high *Informative* and high *Correlation Coefficient*. On the other hand, although the concept `Person` is still a quite an important concept (ranking 8th), it has a much higher TF value but a lower *Semantic Impact* value when compared with `Date`. This is due to its relatively small *Informative Coefficient* value even if there is a strong correlation with the other concepts. Intuitively, this is correct because all the news articles in the corpora are about Donald Trump and therefore the concept of `Person` may not as general as the other concepts with a higher *Semantic Impact* value which leads to a small *Informative Coefficient* value as the result show.

Table 4. Top 10 class pairs in the Source Corpus

X var	**Y var**	**MIC (strength)**
Organisation	Place	0.64962
Date	Event	0.64915
Event	Organisation	0.62966

Facility	Organisation	0.60051
Cardinal	Event	0.60044
Facility	Place	0.57699
Cardinal	Organisation	0.56309
EventPersonnel	Event	0.54543
EventPersonnel	Date	0.54005
Cardinal	Date	0.52521

Table 5. Correlation Coefficient and Semantic Impact (sorted by Semantic Impact)

Concept/Class	**Correlation Coefficient**	**Semantic Impact**
Event	0.592085436	1.360595612
Date	0.539361512	1.197490259
Organisation	0.517221219	1.107800123
Place	0.470012372	1.031985304
Cardinal	0.466077514	0.964520852
EventPersonnel	0.474035643	0.793774204
EventPerformance	0.369981821	0.791260849
Person	0.431227123	0.756322023
EventViolence	0.343677231	0.750650086
Facility	0.474322953	0.505266126
EventElection	0.385609452	0.43937166
EventCustody	0.141969431	0.348065975
Crime	0.349006108	0.293063856
Product	0.137378303	0.211302975
EventBusiness	-0.007672398	0.19819171
SportingEvent	-0.029443302	0.054561293
Award	0.026273671	0.032015782
Time	0.068046355	0.014121219
TitleWork	0.061481863	0.013583167
HealthCondition	-6.18E-04	0.010321998
Vehicle	-0.011557869	-0.016295982
EntertainmentAward	-0.108546672	-0.056908001
NaturalDisaster	-0.171057624	-0.072654592
GeographicFeature	-0.192902362	-0.138340239
EventDemonstration	-0.180262855	-0.153971247
Weapon	-0.285569086	-0.218614866
EventEducation	-0.170707944	-0.25731984
Duration	-0.340206431	-0.334569616
Substance	-0.414047743	-0.432988886

4 Conclusion

In order to measure the importance of a particular concept to the domain knowledge at the semantic level, this paper introduced a new idea called the “Semantic Impact” which is computed from a concept’s *Informative Coefficient* and *Correlation Coefficient*.

In Section 2, we explained the method and the process to calculate these coefficients for domain concepts. We evaluated this by using 200 BBC News articles on Donald Trump and discussed the results in Section 3.

We have also briefly analysed the preliminary evaluation result and explained why `Event`, `Date` and `Organisation` have a higher *Semantic Impact* value over the others. Specifically, the concept `Person` is used as an example to explain why a high *Term Frequency* value may not necessarily result in a high *Semantic Impact* value. At this stage, we can mainly assess these results intuitively. A quantitative evaluation will be required to apply our semantic impact measure and the computation approach to other domains. This is also the crux of the future work for this research.

References

1. Zúñiga, G.L., *Ontology: its transformation from philosophy to information systems*, in *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*. 2001, ACM: Ogunquit, Maine, USA. p. 187-197.
2. Maedche, A. and S.J. Staab, *Ontology learning for the Semantic Web*. 2001. **16**(2): p. 72-79.
3. Wong, W., W. Liu, and M. Bennamoun, *Ontology Learning from Text: A Look Back and into the Future.* ACM Computing Surveys, 2012. **44**(4): p. 1-36.
4. Horrocks, I., *Ontologies and the Semantic Web.* Communications of the ACM, 2008. **51**(12): p. 58-67.
5. Beel, J., C. Breitinger, and S.J. Langer, *Evaluating the CC-IDF citation-weighting scheme: how effectively can 'Inverse Document Frequency'(IDF) be applied to references.* Proceedings of the 12th iConference, 2017.
6. Wan, J. and J. Barnden, *A New Semantic Model for Domain-Ontology Learning*, in *Human-Centered Computing*. 2015, Springer Lecture Notes in Computer Science: Cambodia. p. 140 - 155.
7. Evert, S., *Distributional semantic models.* NAACL HLT 2010 Tutorial Abstracts, 2010: p. 15-18.
8. Mikolov, T., et al., *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781, 2013.
9. Mikolov, T., et al., *Distributed Representations of Words and Phrases and their Compositionality*, in *Advances in Neural INformation Processing Systems 26 (NIPS 2013)*. 2013. p. 3111--3119.
10. Reshef, D.N., et al., *Detecting novel associations in large data sets.* Science, 2011. **334**(6062): p. 1518-1524.
11. IBM. *Natural Language Understanding*. Watson Developer Cloud 2017 [cited 2017; Available from: https://www.ibm.com/watson/developercloud/natural-language-understanding.html.
12. BBC. *Trump says terror attacks 'under-reported': Is that true?* US & Canada 2017 [cited 2017 10/02/2017]; Available from: http://www.bbc.co.uk/news/world-us-canada-38890090.

13. BBC. *Core Concepts Ontology*. 2015 01/04/2018]; Available from: https://www.bbc.co.uk/ontologies/coreconcepts.
14. Deeplearning4J. *Troubleshooting & Tuning Word2Vec*. Word2Vec 2017 [cited 2017 1st Feb]; Available from: https://deeplearning4j.org/word2vec#trouble.
15. Glorot, X. and Y. Bengio. *Understanding the difficulty of training deep feedforward neural networks*. in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010.
16. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.