JOURNAL OF INNOVATION IN DIGITAL ECOSYSTEMS 3 (2016) 141–147

HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jides

Meaning-based machine learning for information assurance

*Courtney Falk**, *Lauren Stuart*

Purdue University, West Lafayette, IN, United States

HIGHLIGHTS

- Describes the combination of semantic knowledge bases with machine learning.
- Natural language processing application for phishing detection.
- Semantic machine learning improves on existing approaches.

ARTICLE INFO

Article history:
Published online 23 November 2016

Keywords:
Natural language processing
Machine learning
Information security

ABSTRACT

This paper presents meaning-based machine learning, the use of semantically meaningful input data into machine learning systems in order to produce output that is meaningful to a human user where the semantic input comes from the Ontological Semantics Technology theory of natural language processing. How to bridge from knowledge-based natural language processing architectures to traditional machine learning systems is described to include high-level descriptions of the steps taken. These meaning-based machine learning systems are then applied to problems in information assurance and security that remain unsolved and feature large amounts of natural language text.

1. Introduction

This paper outlines a research program called meaning-based machine learning (MBML). MBML combines the meaningful input provided by ontological semantics with the pattern searching abilities of established machine learning.

First, the paper explains the novelty of MBML and establishes how it interconnects with different fields.

Second, the end-to-end data flow of an MBML system is described. Special attention is paid to leveraged established formalisms from ontological semantics.

Finally, there is a discussion of how this general MBML approach is applicable to problems of information assurance. The problems of phishing detection and stylometry are addressed in-depth.

1.1. *Machine learning*

Machine learning (ML), particularly statistical ML, has matured and grown in popularity over the past decade for natural language processing (NLP) applications. Some, but not necessarily all, of the most popular ML approaches

Peer review under responsibility of Qassim University.

* *Corresponding author.*
E-mail address: courtney.falk@gmail.com (C. Falk).

http://dx.doi.org/10.1016/j.jides.2016.10.007

center around statistical techniques [1]. Performance of these statistical methods improve with larger amounts of well-annotated data.

Different ML approaches attempt delve below surface language features such as word frequency and syntactic structure into semantic meaning with varying levels of success. Whether or not statistical approaches can identify semantic information remains an open question that is outside the scope of this paper. Instead, the MBML approach described in detail later on will start from the position of using semantically meaningful data derived from an ontological semantics system. It is the position of the authors that only by beginning with semantic data as the input will the output resemble anything approaching what humans understand to be semantically meaningful.

It is always worth noting that the sense in which the aforementioned statistical ML systems use the word "semantics" differs from the "semantics" of ontological semantics. In the former sense "semantics" describes a structure that is sufficiently complex to example the observed data while in the latter sense "semantics" describes the philosophical, linguistic, and cognitive models of meaning.

1.2. *Ontological semantics technology*

Ontological Semantics Technology (OST) is a development of the theory of Ontological Semantics [2]. Ontological Semantics first began to be formalized as a comprehensive system with the Mikrokosmos project [3] before it was described in detail in the text of the same name [4].

At its core, ontological semantics is a frame-based system [5] where language-dependent lexicons define syntactic behavior and extend the semantic concepts stored in the language-independent ontology. The development of these resources (the lexicons, other language-specific knowledge repositories or tools, the ontology, and other language-independent knowledge repositories or tools) is named acquisition; its practitioners are acquirers [4].

The process of acquisition involves the careful description of linguistic-semantic behaviors and distinctions, as observed or theorized in human use of language, via the OST framework. The two basic resources, the lexicon and the ontology, are the two we will discuss in depth here because the details of their specification and intended use most impact the array of features we wish to introduce. Other elements in the ecology of OST are described elsewhere.

The ontology is a large, dense graph of nodes, called concepts, connected by relations. A concept represents a separable, cohesive meaning unit, such as *automobile*, *travel*, *rice*, or *freedom*. Relations provide relative information for concepts; they have a domain (originating concept), and range (target concept, literal, or scalar) by which additional information is encoded. The strength of an ontology is in its dense connections between concepts: the use of a *automobile* for a *human* in an instance of *travel* is modeled by appropriately-restricted (loose enough to make semantic distinctions where actual text does, but tight enough to reduce sense-making where actual text would not) relations (where *human* is the *AGENT* of *travel*), along which some very basic reasoning can be performed. The methods and directions of such reasoning become application-specific (for instance, in detecting and flagging possible instances of insider threat) but OST assumes a reusable kernel of these, that we also assume here to be in any OST implementation regardless of application.

The lexicon provides the first mapping from word (or other separable part of a text or utterance) to concept, relation, attribute, or graph of these. A lexicon entry gives, for each sense of a word, the base lexeme, morphological rules, syntactic and grammar rules and representation, and semantic representation. This semantic representation specifies the ontological concepts, relations, or literals that express the meaning of the lexeme. In text processing, each word (or phrasal set of words, in the case of common multiple-word expressions with non-compositional semantics) is queried in the lexicon, which gives one or several sets of morphological, syntactic, and semantic dependencies to be resolved in assembling the semantic map of the text's meaning. (Some special cases may be handled instead by other lookup-type elements of OST; for example, proper names are stored in a separate resource, the onomasticon, and have some other considerations for how they show up in the map.)

OST processes a text into TMRs, text meaning representations. A TMR constitutes a modified subgraph of the ontology, encoding information that has been explicitly or implicitly called out in the text. The granularity is application-determined: some applications may find that a one-to-one sentence-to-TMR transformation is all that is needed or can be done with what is available, and some may operate on a whole text and produce one large and complicated TMR. It is this graph of concepts, relations, and literals that we use as the input for MBML.

1.3. *Information assurance and security*

Information assurance and security (IAS) are ripe fields for NLP applications [6–8]. Because natural language remains an unsolved problem and yet is central to how humans use technology it is an important research area for IAS.

Semantically meaningful results in NLP can offer new insight into text-heavy domains such as social network analysis, business intelligence, and social engineering detection. As in [7], we use our Section 3 to explore a few problem areas in information assurance and security in which we have noted a need.

1.4. *What is meaning-based machine learning?*

MBML bridges disciplines. It begins in the realm of ontological semantics and uses techniques popularized by machine learning (ML) to find patterns in meaningful data. For an MBML system that relies on OST the meaning is represented in the TMRs. ML techniques examining these meaningful TMRs will in turn derive meaningful results from the TMRs.

The kinds of patterns in TMRs varies. Different linguistic phenomena aren't necessarily represented solely in the text itself. Novelty of information and referencing information across documents assume a certain level of background knowledge. It is in areas such as these that ML algorithms, operating on the TMR structures generated by OST, that ML might add new layers of meaning by building on the existing meaning described by OST.

2. Data flow

MBML advocates the use of meaning representations as a source of features for machine learning with text; this section explores how TMRs may be used.

As a meaning representation, a TMR is a graph of meaning entities (concepts) connected by meaningful edges (properties). These graphs can be decomposed into subgraphs for the creation of feature vectors in a number of ways; the following list is not exhaustive, but rather is a foundation from which to build.

2.1. *Concept or relation names*

A family of features can be defined over the occurrences of concept or relation names, the analogue to word vectors in text processing. For instance, a frequency analysis of concept and relation names may differentiate texts with different topics. A text might also be characterized by relative frequencies of related or contrasting concepts (does a text refer, more often than another text, to the event concept covering the act of eating rather than that covering drinking?) or relations (does a text call out, more often than another text, the *AGENT* relation of events rather than *LOCATION*s?).

To distinguish between particular instances of a concept in the TMRs as written here (e.g.: a text refers to two separate cars), the concept-names have numbers appended in order to create unique identifiers. By "concept name" we mean the name of the concept; in the TMRs that appear here, this is the portion of the node name that precedes the hyphen.

2.2. *Concept families*

The hierarchical nature of some ontological relations (more on this in point 3 in the next subsection) reflects a scale of generality and specificity that can be treated as a slider in detail level. Sets of features can be defined in terms of the topmost (least specific) concept that should be considered, or in the maximum depth of specificity. The analysis may be closed down to families of concepts that inherit from a certain concept (e.g.: consider all of the children of *vehicle*, which includes *aircraft*, *yacht*, and *honda-civic*) or closed up from a certain level of children (e.g.: consider concepts no more specific than *automobile* so as not to differentiate between *honda-civic* and *dodge-dart*, or consider children only above a depth of *n* from the root).

2.3. *Relation families*

OST distinguishes between several types of relations. One major source of distinction is in argument count and type; another is in the nature of the relationship that the relation encodes.

2.3.1. *Range families*

A relation with a concept range is a property; properties connect two (or more) concepts. A relation with a literal range is an attribute; an attribute is a detail of the concept that does not concern other concepts. If a relation is expressed in a TMR, a value in its range is selected; we call it the filler. OST also considers different facets for relation ranges – one is SEM, which provides selectional restrictions on the filler – but we will consider only VALUE, the facet that expresses what the actual filler is for the TMR, for our discussion of OST and ML, as it is the most common facet in use in TMRs [4, p. 199].

2.3.2. *Argument-count families*

To date, the properties defined in OST have all been two-place, but some meaningful relationships between concepts may be better expressed as *n*-ary relations of a higher *n*. At the level of notation, this distinction may not be functionally useful: any *n*-ary relation may be expressed as a set of binary relations; however, the decision to acquire, and represent meaning with, any non-binary relation is indicative of a distinction that should also be taken note of in any processing of a TMR. The exact representation of these TMRs will affect the creation of features based in property names, but only as much as any other evolution or tweak in the language used to write TMRs. The inclusion of non-binary relations as a separate feature or family of features may be useful when those relations are, for example, indicative of some other level of complexity or detail in the text being processed.

2.3.3. *Meaning/function families*

A subset of properties, called taxonomic, comprise the usual backbone of ontologies: the parent–child/superclass–subclass relationships. Taxonomic properties serve mostly to provide hierarchical structure in the ontology, providing family trees for reasoning along inheritance or mereological lines, but they may appear in TMRs if evoked in the source text. For example, an introductory text giving background information on a topic might reasonably be expected to contain some sentences like *x is a type of y* or *x comprises y, z, etc.*

Another subset of properties represents thematic roles, such as subject, agent, and beneficiary. These can be considered shorthand for syntactic structures? training on this subset of properties may reveal more about the surface characteristics of a text. The appearance of a thematic role property in a TMR may reflect a lack of detail required to further disambiguate the text; for instance, the relation of one concept to another with only the *AGENT* property might elide a more expressive, precise relationship. There are several reasons that a relatively imprecise property could appear that do not have much to do with the source text: if the static resources do not capture any more precise relationship between two entities, then there is an acquisition gap; if the ability to represent the relationship is there, there may be a fault in processing or a lack of information in the source text that would otherwise enable the processing to push the specificity of the TMR to that level.

2.4. *Denormalization*

The next, more complex, unit of meaning of a TMR is a ⟨concept, relation, filler⟩ triple: the combination of two concepts (or a concept and a literal) and the way in which

```
(BUY–13
   (AGENT (VALUE (HUMAN–117
      (HAS–NAME (VALUE (GIVEN–NAME–4)))
   )))
   (THEME (VALUE (CAR–312
      (HAS–COLOR (VALUE (BLUE)))
   )))
)
```

Fig. 1 – An example proposition.

they are related. Any TMR can be specified as a list of such triples. Denormalizing the static knowledge structures into OST isn't an entirely novel concept. Earlier work by Taylor [9] used denormalized structure triples in storing the structures in a database. This idea harkens back to the triple stores favored by Resource Description Framework (RDF) featured as a part of the semantic web [10]. This paper differs from the previous work in the function the tuples serve. Instead of being a mechanism for storing complex data structures, the tuples are used as discrete machine learning features.

As mentioned before, with consideration of the full range of facet types, these triples are actually quads (variations in ⟨concept, relation, facet, filler⟩); however, we focus here on TMRs with VALUE facets, so quads are reduced to triples. The below example shows a sentence, a TMR for that sentence, and some example triples derived from the TMR. Note that though there is a single head fact in *buy*-1, the denormalization produces two triples.

Let's demonstrate using a very simple sentence as an example:

1. "John buys a blue car".

The five words of Example 1 generate the proposition tree described via s-expression in Fig. 1:

The nested properties described by the s-expression in Fig. 1 hide some of the knowledge gained from parsing the example sentence. These proposition trees are a parsimonious way of representing the knowledge produced. However, in OST, relations are allowed inverses such that the domain of one relation becomes the range of the inverse relation and vice versa. Eq. (1) below succinctly expresses that logic. What this means for translating proposition trees into tuples is that not only do the tuples that are explicitly described in the proposition tree require handling, but so do any inverse relations.

$$(\forall p \in R)(\exists q \in R)(Inverse(p, q) \iff Inverse(q, p)). \quad (1)$$

Feng et al. [11] employ a similar technique in how they decompose syntactic parse trees into discrete features. Since propositions and TMRs present similar structures to those trees we adapt their approach to generate our features.

Generating features from the tree-like s-expression seen in Fig. 1 is a simple matter of performing a depth-first tree traversal. So once the TMR is produced the subsequent step of generating denormalized feature for ML processors runs in $O(n)$ time.

The final result is a set of seven triples describing all knowledge gained from parsing the five-word example sentence. In the ontology used for this example, the HAS-COLOR property is an attribute, not a relation, therefore it doesn't have an inverse property.

1. ⟨BUY-13, AGENT, HUMAN-117⟩
2. ⟨HUMAN-117, AGENT-OF, BUY-13⟩
3. ⟨HUMAN-117, HAS-NAME, GIVEN-NAME-4⟩
4. ⟨GIVEN-NAME-4, IS-NAME-OF, HUMAN-117⟩
5. ⟨BUY-13, THEME, CAR-312⟩
6. ⟨CAR-312, THEME-OF, BUY-13⟩
7. ⟨CAR-312, HAS-COLOR, BLUE⟩.

Triples as proposed above provide a way of learning based on purely semantic structures, but with more expressive potential than the property-names set of features. The triple as a minimal meaningful subgraph is analogous to a trigram (see Table 1).

2.5. *Subgraphs and [sub]TMRs*

Features may be derived from the structures of the graphs obtained as well as from the structures of the TMR as a whole: connectedness, depth, and other measures of complexity may be useful in characterizing texts via the characteristics of their TMRs, and the same is true for subgraphs of those TMRs, however they are obtained.

One may decompose a TMR graph into subgraphs connected only by a particular relation. The process outlined in [9] is proposed for ontology verification, but has utility in sectioning large TMRs for analysis of chains and components. This is similar to denormalizing the whole TMR, as proposed in the immediately previous subsection, and focusing only on triples with a particular relation.

Likewise, the number, complexity, and nature of TMR branches for instances of a particular concept may be of interest. Finally, particular subgraphs may be sources of characteristics for TMRs: the number, frequency, or context in which a particular fact, event, or object is referred to (or implicitly called out, or obliquely represented) may be of interest as a feature.

2.6. *Surface-to-structure mapping*

As a text is processed in OST, its range of potential meanings is narrowed—from a purely combinatorial analysis, the number of possible meanings is exponential in the number of words, and the process of attempting to fit these together with the selectional restrictions imposed upon them through information in the lexeme entries and the ontology knocks a large number of these out of consideration. As such, the mapping from surface form to deep structure would be of interest as either a feature or a hypothesis. Such a feature would appear as a duple, where the first position is the string representation of the root of the lexeme and the second place is the concept that it maps to. To continue with the example sentence from before we give the duples "bought" ⇒ ⟨"buy", PURCHASE⟩ and "car" ⇒ ⟨"car", AUTOMOBILE⟩. These duples offer a second type of feature that can help an algorithm learn about the significance of the mappings from surface structure to those of deep meaning structures.

Table 1 – The different types of features allowed in regards to single (upper) and triple (lower) tuples.

	Unlexicalized	Lexicalized
Concept	⟨HUMAN⟩	⟨CAR, "automobile"⟩
Fact	⟨CAR-1⟩	⟨HUMAN-1, "Joe"⟩
Concept	⟨CAR, HAS-COLOR, BLUE⟩	⟨BUY, AGENT, HUMAN, "buy"⟩
Fact	⟨CAR-1, THEME-OF, BUY-1⟩	⟨WAR-1, AGENT, NATION-1, "WW2"⟩

3. Applications

In keeping with the prior work of Raskin et al. [7], we examine problems in IAS to see how MBML might provide solutions. Increasing reliance on computer system for critical infrastructure, commerce, and governance means that IAS is more important now than ever.

3.1. *Phishing detection*

Phishing detection presents unique opportunities for NLP applications. The content of the phishing email is critical to the success of the phishing attack. A successful phishing email convinces the recipient to complete the attack on the behalf of the attacker. An NLP-based approach to detecting phishing emails could prove more generalizable and robust than prior approaches that are based on meta-data features of the phishing emails.

Phishing detection approaches that depend on meta-data extracted from the email are brittle. These features could include MIME headers, details about the URLs used in the message, or the domain of the email sender. Identifying these kinds of features is technologically simpler because it can be done regular expressions or other, easily computed techniques. The downside is that these features are very quickly changed by the attacker. An attacker sending phishing emails can utilize a distributed botnet with a different IP address and headers in every few messages.

What attackers cannot rapidly change for large number of targets is the content of the phishing email. An attacker will spend time crafting the message to make it effective for a wide range of readers and then use it in a concerted phishing campaign that sees the message sent out to hundreds or thousands of recipients. A generalized technique of identifying phishing emails based on the content of the message could render ineffective entire campaigns instead of single messages.

Preliminary results of experiments utilizing the MBML methods described in this paper are favorable. In comparing binary classifier machine learning algorithm performance between text strings and TMR triples, the MBML approach performed than the corresponding unigram model classifier when using naive Bayes and J48 algorithms [12]. Fig. 2 compares the F_1 scores of both unigram and TMR language models by which algorithm was used for the classifier.

Falk's work is preliminary and features a small sample size. One lingering question was whether or not the results seen were significant or not. Table 2 shows the results of taking the data seen by Falk and running it through an effect size calculator [13]. According to the Cohen's d, the naive Bayes classifier show a medium-to-large effect while the J48 decision tree classifier shows a large effect size.

Table 2 – Effect sizes of the naive Bayes and J48 classifiers seen by Falk.

Algorithm	Cohen's *d*	Effect size *r*
Naive Bayes	0.785	0.365
SVM	0.229	0.114
J48	1.130	0.492

These results provide a promising initial study into whether or not OST-based MBML is a viable way to improve upon existing NLP ML approaches.

3.2. *Stylometry and authorship attribution*

The field of stylometry attempts to quantify and measure an author's writing style, in support of making, evaluating, and/or supporting claims of authorship. Recent acceleration in the advancement of the field reflects an increasing impulse, and lagging capability, to automate and scale author recognition. A highly useful metric is word choice: the author's selection of a particular way to express an idea, in the face of a range of available ways, is deemed a reliable and measurable way to characterize the author. The feature footprint of this intuition has been pursued in word vectors at large and in the definition of synonym sets, but with some level of semantic analysis comes a way to expand the lens from individual words (and ideas that are expressible in individual words) to much larger windows. The translation of a natural language text into a language-independent interlingua (here, TMRs) renders the variability of expressions into a restricted range—there are many ways to talk about John's purchase of a car, but the TMR should always contain the event PURCHASE and the object AUTOMOBILE. The way that the author chooses to represent that event in the text is potentially unique or recognizable, so the mapping from surface to deep structure would be useful for characterization.

OST researchers are actively investigating how knowledge-based approaches affect the performance of stylometry [14]. This early work by Hinh et al. [14] leverages the large resources available in FrameNet [15]. FrameNet, while different in formalism from OST, provides enough similarities between the two to allow for initial investigations that are applicable to both theories.

In general, the addition of semantic information as a domain from which to draw features allows for more space in which to capture the variability and similarity of authors. Other hallmarks of TMRs could also be understood as hallmarks of authors, or TMR information, considered in conjunction with other sources of information, could paint a more expressive picture of an author's idiosyncrasies in writing.

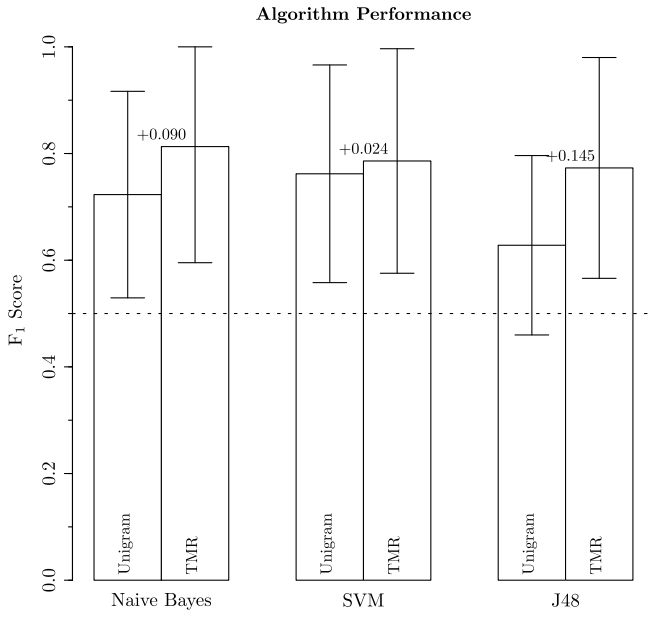

Fig. 2 – F-scores of the machine learning models grouped by training data set.

3.3. *Generalizing*

Any data is potentially expressible in the language of the TMR; though OST was conceived for the understanding of natural language, any of its reasoning modules may operate on data of any kind that has been translated into TMRs. Any machine learning task that deals with or requires some meaningful data could be done with that data translated into TMRs and analyzed in the directions laid out in Section 2 here. The translation of both text and non-text data into the same interlingua for reasoning and analysis that is agnostic of the origin and original form of that data is a tempting possible state of affairs in any application, though there are easy analogues in summarization (the transformation of many TMRs, perhaps from network traffic, into natural language digests for human consumption) and in stylometry/attribution (that same network traffic, analyzed for the fingerprints of network attacks and attackers).

4. Bridging across semantics

One of the shortcomings of an OST approach is its reliance on a knowledge base that is acquired by hand. Manually acquired knowledge is slow and expensive to accumulate. Other NLP and ML approaches exist that also invoke the label of semantics. Latent semantic analysis (LSA) [16] and latent Direchlet allocation (LDA) [17] compare corpora of documents to find concepts and topics respectively. The difference in how the term "semantics" is used in OST as compared to LSA or LDA is that in OST the meaning comes first and is explicitly described by a human while for the latter two approaches the actual meaning of the vectors isn't apparent until they are examined by a human after the processing is complete. The machine processing the corpora isn't the one that is determining what a meaningful result is.

Despite the difference in how they approach semantics, LSA and LDA feature automatic processing to reach what they call semantics. An OST system would benefit greatly from such an automated acquisition process. A tangential research question in the MBML research program is whether or not an LSA or LDA system might be harnessed in such a way as the output vectors are then mapped onto the knowledge resources of an OST system?

The idea of mapping from a vector space to an ontology is not a new idea. Chen et al. [18] describe utilizing LSA as a way of generating new concepts that are leaf nodes of an existing ontology. This idea naturally works with both domain ontologies and more abstract upper ontologies [19]. Banjade et al. [20] also studied LSA but their work differed from Chen et al. [18] in that they trained an artificial neural network to map between two different vector spaces.

Now we suggest the outline of what an LSA-to-OST mapping algorithm might look like. The first step is to run a corpus of documents through an LSA processor. Second, take the terms (lexemes) from the LSA vector space and identify which of these are already represented in the lexicon and ontology. Third, take all unrepresented terms and find their nearest neighbor term that is indeed represented. Finally, create a new concept for the unrepresented term and connect it to the nearest represented term's concept via the appropriate kind of property.

The above approach is not without its complications. Polysemy is when a single lexeme has multiple different meanings or senses. Our outline doesn't address this and instead assumes a single sense per term. The second problem is how to determine which kind of property is appropriate for connecting the new concept to the existing concept.

For instance, if we have the terms "dog" and "Doberman" when "dog" is already represented. In this case it would require a taxonomic IS-A from "Doberman" to the more abstract "dog" concept. Another example might be when

terms “harvest” and “quinoa” co-occur when “harvest” is already known. In this case a taxonomic relationship is incorrect because “harvest” is an agricultural event while “quinoa” is the theme or product of “harvest”. This would require a case role or other type of semantic relation as the appropriate property.

A short-term solution to picking properties would be to have a human acquirer perform the task. While this still slower than a fully automated acquisition system it would still be faster than a human performing the entire task manually. This might be more accurately described as augmented human acquisition. The long-term implementation goal of any OST system should still be a fully automated acquisition process.

5. Conclusion

This paper outlined MBML as a novel way of combining ontological semantics with machine learning. The machine learning algorithms find patterns in the meaningful input data. A proposed end-to-end data flow described how the OST input becomes ML output. A successful MBML system would perform superior to ML approaches that rely only on shallow surface or syntactic features for language modeling. The benefits of an MBML system extend to several areas of information assurance including, but not necessarily limited to, phishing detection, stylometry, and astroturfing detection. Because MBML is built around the knowledge-based OST theory of NLP, the resources constructed for one application can be quickly transferred to another, new application. All of these design aspects make MBML a ML system able to apply human understandable meaning across a general range of applications.

REFERENCES

[1] S.J. Russell, P. Norving, Artificial Intelligence: A Modern Approach, second ed., Pearson Education, 2003.

[2] J. Taylor, Computational semantic detection of information overlap in text, in: Proceedings of Cognitive Science Conference, 2010.

[3] B. Onyshkevich, S. Nirenburg, A lexicon for knowledge-based mt, Mach. Transl. 10 (1995) 5–57.

[4] S. Nirenburg, V. Raskin, Ontological Semantics, MIT Press, 2004.

[5] J. Sowa, Knowledge Representation: Logical, Philosophical and Computational Foundations, Brooks/Cole, 2000.

[6] V. Raskin, C.F. Hempelmann, K.E. Triezenberg, S. Nirenburg, Ontology in information security: a useful theoretical foundation and methodological tool, in: Proceedings of the 2001 Workshop on New Security Paradigms, ACM, 2001, pp. 53–59.

[7] V. Raskin, S. Nirenburg, M.J. Atallah, C.F. Hempelmann, K.E. Trizenberg, Why NLP should move into IAS, in: Proceedings of the 2002 COLING workshop, Vol. 13, 2002, pp. 1–7.

[8] V. Raskin, J. Taylor, A Fresh Look at Semantic Natural Language Information Assurance and Security: NL IAS from Watermarking and Downgrading to Discovering Unintended Inferences and Situational Conceptual Defaults, Elsevier, 2013, pp. 45–61.

[9] J. Taylor, V. Raskin, Graph decomposition and its use for ontology verification and semantic representation, in: Intelligent Linguistic Technologies Workshop at International Conference on Artificial Intelligence, 2011.

[10] World Wide Web Consortium, Rdf 1.1 concepts and abstract syntax, 2 2015. URL http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[11] S. Feng, R. Banerjee, Y. Choi, Syntactic stylometry for deception detection, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Vol. 2, 2012, pp. 171–175.

[12] C. Falk, Knowledge modelling of phishing emails (Ph.D. thesis), Purdue University, West Lafayette, 2016.

[13] L. Becker, Effect size calculators, Online, March 2000. URL http://www.uccs.edu/~lbecker/.

[14] R. Hinh, S. Shin, J. Taylor, Using frame semantics in authorship attribution, final report produced for a course, 2016.

[15] C.F. Baker, C.J. Fillmore, J.B. Lowe, The Berkeley FrameNet project, in: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1, ACL’98, Association for Computational Linguistics, Stroudsburg, PA, USA, 1998, pp. 86–90. http://dx.doi.org/10.3115/980845.980860. URL http://dx.doi.org.ezproxy.lib.purdue.edu/10.3115/980845.980860.

[16] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Amer. Soc. Inf. Sci. 41 (6) (1990) 391.

[17] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan.) (2003) 993–1022.

[18] R. Chen, Y. Lee, R. Pan, Adding new concepts on the domain ontology based on semantic similarity, in: International Conference on Business and information, 2006, pp. 12–14.

[19] V. Nguyen, Ontologies and information systems: a literature survey, Tech. Rep., Defence Science and Technology Organisation, 2011.

[20] R. Banjade, N. Maharjan, D. Gautam, V. Rus, Handling missing words by mapping across word vector representations, in: Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, 2016.