

# Meaning-Based Machine Learning for Information Assurance

**Courtney Falk and Lauren Stuart**

Computer and Information Technology

Purdue University

West Lafayette, Indiana 47907

{falkc, lstuart}@purdue.edu

## Abstract

This paper presents meaning-based machine learning (MBML), the use of semantic input into machine learning systems in order to gain meaningful output. The semantic input comes from the ontological semantics theory of natural language processing. Machine learning enables the finding of patterns within this semantic data. Applications for MBML in the area of information assurance include phishing detection, stylometry, and other problems examined in prior ontological semantics research.

## 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.

## 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 center around statistical techniques (Russell and Norving 2003). 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.

## Ontological Semantics Technology

Ontological Semantics Technology (OST) is a recent branch of the field of ontological semantics (Taylor 2010). Ontological semantics got its start with the Mikrokosmos project (Onyshkevich and Nirenburg 1995) before it was formalized in the text of the same name (Nirenburg and Raskin 2004).

At its core, ontological semantics is a frame-based system (Sowa 2000) 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 (Nirenburg and Raskin 2004).

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.

## Information Assurance and Security

Information assurance and security (IAS) are ripe fields for NLP applications as noted by Raskin et al. (Raskin et al. 2002). Natural language remains an unsolved problem for computational approaches.

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 (Raskin et al. 2002), we use our Section III to explore a few problem areas in information assurance and security in which we have noted a need

## 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 mean-

ingful 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.

## 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 sub-graphs 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.

## 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 *LOCATIONS*?).

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.

## 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).

## 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.

**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 (p. 199, (Nirenburg and Raskin 2004)).

**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.

**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.

## Denormalization

The next, more complex, unit of meaning of a TMR is a  $\langle \text{concept, relation, filler} \rangle$  triple: the combination of two concepts (or a concept and a literal) and the way in which 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 in and of itself. Earlier work used denormalized structure triples in storing the structures in a database (Taylor and Raskin 2011). This idea harkens back to the triple stores favored by Resource Description Framework (RDF) featured as a part of the semantic web (World Wide Web Consortium 2015). 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  $\langle \text{concept, relation, facet, filler} \rangle$ ); 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 Figure 1 below:

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

Figure 1: An example proposition.

The nested properties described by the s-expression in Figure 1 hide some of the knowledge gained from parsing the example sentence. These proposition trees are a parsimonious way of representing the knowledge produced. But in ontological semantics relation properties have inverses such that the root of one relation becomes the filler of the inverse relation and vice versa. Equation 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.

$$(\forall p \in R)(\exists q \in R)(\text{Inverse}(p, q) \iff \text{Inverse}(q, p)) \quad (1)$$

Feng et al. (Feng, Banerjee, and Choi 2012) 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.

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.

### 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 (Taylor and Raskin 2011) 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.

### 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”  $\Rightarrow$  <“buy”, *purchase*> and “car”  $\Rightarrow$  <“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.

## Applications

In keeping with the prior work of Raskin et al. (Raskin et al. 2002), 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.

### 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 at least as well if not better in every scenario (Falk 2016).

### 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

Table 1: The different types of features allowed.

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

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.

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.

## 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 II 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).

## 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. The benefits of an

MBML system extend to several areas of information assurance including, but not necessarily limited to, phishing detection and stylometry.

## References

- Falk, C. 2016. Identifying phishing emails by their meaning.
- Feng, S.; Banerjee, R.; and Choi, Y. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers*, volume 2, 171–175.
- Nirenburg, S., and Raskin, V. 2004. *Ontological Semantics*. MIT Press.
- Onyshkevich, B., and Nirenburg, S. 1995. A lexicon for knowledge-based mt. 10:5–57.
- Raskin, V.; Nirenburg, S.; Atallah, M. J.; Hempelmann, C. F.; and Trizenberg, K. E. 2002. Why nlp should move into ias. In *Proceedings of the 2002 COLING workshop*, volume 13, 1–7.
- Russell, S. J., and Norving, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition.
- Sowa, J. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole.
- Taylor, J., and Raskin, V. 2011. Graph decomposition and its use for ontology verification and semantic representation. In *Intelligent Linguistic Technologies Workshop at International Conference on Artificial Intelligence*.
- Taylor, J. 2010. Computational semantic detection of information overlap in text. In *Proceedings of Cognitive Science Conference*.
- World Wide Web Consortium. 2015. Rdf 1.1 concepts and abstract syntax.