Informatics4AI Whitepaper

The current state of:

Data Labeling Full-Text Datasets for AI Predictive Lift.

			
Fox	Turtle	Tiger	Frog

To paraphrase James Kobielus (SiliconANGLE), it is well understood by data scientists that without high-quality labeled training data, supervised learning falls apart and there is no way to ensure that models can predict with any accuracy. But with full-text datasets, high quality, fully labeled training sets are hard to come by. As if that isn't bad enough, it is also unclear what a high quality, fully labeled full-text dataset looks like. We all know what a labeled dataset looks like for image recognition, it basically consists of two columns: the image, and the label (see the table above).

But with full-text data there are no acknowledged standards or best practices for data labeling. At Informatics4AI, most of our customers create the labeled training data from their own raw unlabeled text. In this whitepaper we examine the current state of data labeling for full-text datasets such as doctors' notes, human conversations, tweets, etc.

We will use the following sentences in this examination:

A. "The x-ray showed clear evidence of a tumor."
B. "The x-ray showed scant evidence of a tumor."
C. "The image was clean."

Humans are currently the gold standard in understanding the world, and they remain the gold standard for labeling data, including full-text. When a person reads the sentences above they understand what they mean. In summary, the sentences B & C mean "the diagnostic image is good news – no cancer," and sentence A means "the diagnostic image is bad news – the patient has cancer." If humans are the gold standard, then why not just use them all the time? Because they are expensive and cannot handle the volumes often required by AI models (or work in real time). Therefore, data scientists are looking for automated methods to transform raw full-text into datasets that a machine can understand with as much accuracy as a human.

Cleaning the data is often a necessary first step. A good list of cleaning techniques to consider can be found here (from Emmanuel Ameisen, AI Lead at Insight AI). These include: tokenization, lemmatization, removal of 'non-words' such as URLs, etc. Once the data is clean, it should be labeled.

This whitepaper will examine the following NLP techniques to label full-text:

- Basic techniques
    - POS Tagging, Chunking, and NER
- Intermediate techniques
    - Bag of Words and TF-IDF
    - Word Embeddings
- Advanced techniques
    - Labeling with an Entity Ontology & for Sentiment
- Future techniques

If you are already familiar with the more well-known labeling techniques discussed below, please jump ahead to the section of most interest to you.

Parts of Speech (POS) Tagging, Chunking, and NER

POS tagging labels words based on how they are used in a sentence. As Jocelyn D'Souza succinctly states, "Chunking works on top of POS tagging, it uses POS-tags as input and provides chunks as output. Similar to POS tags, there are a standard set of Chunk tags like Noun Phrase (NP), Verb Phrase (VP), etc. Chunking is very important when you want to extract information from text such as Locations, Person Names etc." This is referred to as Named Entity Recognition (NER). NER is often critical, as facts and knowledge are normally expressed by named entities (e.g. Who, Where, When, Which), where much of the meaning resides. The following sentence demonstrates this well: "Jeff Bezos, the CEO of Amazon, and the world's richest man, attended Princeton, and purchased the Washington Post in 2013 for $250 million.)

Let's take a look at our POS tagged sample sentences.

Sentence A

Text	The	x-ray	showed	clear evidence	of	a	tumor.
POS		noun	verb	noun phrase	preposition		noun

Sentence B

Text	The	x-ray	showed	scant evidence	of	a	tumor.
POS		noun	verb	noun phrase	preposition		noun

Sentence C

Text	The	Image	is	clean.
POS		Noun	verb	adjective

The POS tagging and chunking label the text, and in some domains such as Weather – NER is all that is needed to understand the text. As once you understand the location, the rest is easy (e.g. "What's the temperature in Boston?"). But in the above example, there are no Named Entities, and therefore POS, Chunking and NER would not help a machine understand the meaning of the sentences. It is often the case that full-text has lots of entities (nouns & noun phrases) that are not named entities and thus other methods to label the text for meaning must be used.

Bag of Words and TF-IDF

Bag of Words attempts to derive meaning based on the theory that "you shall know a word by the company it keeps" (John Firth, 1957). In summary Bag of Words represents each word as a vector based on an index of the total vocabulary, keeping track of the number of times words occur in the piece of text, but ignoring the grammar and order of the words. This enables the machine to quickly and efficiently compare two pieces of text. Bag of Words assumes that documents/sentences have similar meaning if they have similar content.

Several techniques can be used in conjunction with Bag of Words to capture more meaning from the document:

- **TF-IDF** (Term Frequency, Inverse Document Frequency) where words are evaluated for frequency of occurrence, discounting words with high frequency (e.g "the") and elevating words that occur rarely (e.g. "Glioblastoma").

- **N-grams** – where the text is evaluated based on groups of words rather than every word. Each group of words is called a "gram" (e.g. two-word pairs, such as "artificial intelligence" or "machine learning," are referred to as bigrams). The idea being that groups of words ("phrases") capture more meaning than a single word.

Bag of Words, TF-IDF and N-grams attempt to label the text for meaning and have proven useful when building relatively simple classification models such as spam detection (e.g. is the email spam or not), however it is our experience at Informatics4AI that these techniques remain inadequate for more complex models where a deeper understanding of the text is needed.

Word Embedding

Word embeddings is like Bag of Words on steroids. Word embeddings encode general semantic relationships (e.g. meaning) within full-text. Word2Vec & GloVe are popular word embedding algorithms. They work by reading massive amounts of text and gain an understanding of which words appear in similar contexts across the dataset. After being trained, the word embedding generates a vector representation for each word in the vocabulary (e.g. Word2Vec can generate a 300 dimension vector – so the vector is capturing a lot of detail about the word and how it is used in context throughout the dataset), with the idea being that words with similar meaning will have similar vectors.

Word embeddings can be learned from your unannotated dataset. The fact that word embeddings do not require expensive labeled data as an input is a key benefit, as they are a way to begin to encode semantic meaning into the raw dataset at low cost. Using pre-trained word embeddings, which are commonly available, makes word embeddings even more attractive. Word embeddings have been a great leap forward in the ability to label text with meaning and have been hugely influential as they have proven themselves to be much more powerful than Bag-of-Words. However, word embeddings have key limitations, such as those noted by Sebastian Ruder "Word2vec and related methods are shallow approaches that trade expressivity for efficiency. Using word embeddings is like initializing a computer vision model with pretrained representations that only encode edges: they will be helpful for many tasks, but they fail to capture higher-level information that might be even more useful … It should thus come as no surprise that NLP models initialized with these shallow representations still require a huge number of examples to achieve good performance." For example, a common objection to Word2vec type word embeddings is that they handle polysemy (words with multiple meanings) poorly. Negation is also often an issue.

Next well take a look at how it's possible to build on word embeddings to improve the accuracy of many models.

Labeling Text with an Entity Ontology and for Sentiment

Two techniques are currently available to improve upon the ability to label text for meaning provided by word embeddings. First, tagging the nouns for meaning with an entity ontology, and second, understanding the micro-sentiment in the text.

Entity Ontologies

An entity ontology is a grouping of words by similar meaning and related concepts (like a thesaurus). The purpose of the entity ontology is to: a) encode commonalities between concepts in a specific domain (e.g. both "yellow fever" and "malaria" are "diseases spread by mosquitoes"), and b) to encode how words relate to concepts, which may vary depending upon the context (e.g. that "hand" is sometimes a worker (hired hand), sometimes applause (give them a big hand), sometimes busy (hands full), and sometimes responsible for (in the hands of the judge). Because the ontology labels the text for meaning based on a more human like understanding of words, at Informatics4AI we have found that it provides better accuracy than imputing meaning based on context/colocation within a dataset (e.g. using bag-of-word or word embedding). Entity ontologies can be created by humans, but more recently they can be created faster and at lower cost in an automated fashion (using AI and building on word embeddings). It is important that entity ontologies be editable by both people and machines so that they can be made more accurate and kept up-to-date.

Several things to note:

- Entity ontologies created in a automated fashion require a large unlabeled dataset (hundreds of thousands or millions of documents), so if you have a small dataset this technique is not useful.
- Some systems create what is known as an orthogonal ontology where nouns and noun phrases are only placed in one concept. While this may be acceptable in some applications, in others it may be highly problematic. (For example, "hand," see above, has multiple meanings and the machine needs to know which one.) Please review the type of ontology being created and how it will be used before you invest heavily in its creation.
- Most ontologies created in an automated fashion have two levels, the concept and the terms (nouns and noun phrases). Ontologies created by humans often have three or more levels, so they can be more precise, but cost much more to build and maintain. We have found that in most situations a two-level non-orthogonal ontology works well for machine learning applications.

Sentiment

Micro-sentiment is also critical when labeling for meaning as sentiment can fully change the meaning of a sentence. And in fact, negation is often very tricky for our clients (and for those using word embeddings), as two sentences can be essentially the same except for one word, yet have opposite meanings.

Many NLP systems claim to enable sentiment detection, but most fall short of the type needed to truly help machines learn in all but the most straightforward of situations. When using sentiment analysis for AI, document-level sentiment is essentially useless. AI requires phrase-level sentiment, and/or entity level sentiment. For example, "The x-ray revealed good news regarding tumor reduction, but also, unfortunately, revealed advanced pneumonia." Machine learning needs to see the first part of the sentence as positive, the second part of the sentence as negative, and also to identify "advanced" as an intensifier and therefore really negative. Lastly, the sentence should not be classified as neutral (half good plus half bad), but rather as two distinct thoughts - one positive and one negative.

Please note that many systems/toolkits come with pre-built sentiment detection capabilities, but Informatics4AI has found that negation and other subtle forms of speech require training to detect and encode accurately. Every domain/dataset is different and the ability to train a system for your specific sentiment requirements is critical.

Let's examine our sample sentences when we label for meaning using an entity ontology and sentiment:

Sentence A

Text	The	x-ray	showed	clear evidence	of	a	tumor.
POS		noun	verb	noun phrase	Pre-position		noun
Meaning		Diagnostic Image					Cancer
Sentiment				Bad sentiment related to tumors			

Sentence B

Text	The	x-ray	showed	scant evidence	of	a	tumor.
POS		noun	verb	noun phrase	Pre-position		noun
Meaning		Diagnostic Image					Cancer
Sentiment				Good sentiment related to tumors			

Sentence C

Text	The	image	is	clean.
POS		noun	verb	adjective
Meaning		Diagnostic Image		
Sentiment				Good sentiment

Having tagged / labeled the data for both meaning and sentiment, the machine has a good chance of understanding the sentences in a manner similar to a human, that sentences B & C mean "the diagnostic image is good news – no cancer," and sentence A means "the diagnostic image is bad news – the patient has cancer."

The Future

Several newer techniques are significantly advancing the state of the art in word embeddings. These new techniques include: ELMo, ULMFiT, and the OpenAI transformer. As per Sebastian Ruder, a paradigm shift is occurring and the industry is "going from just initializing the first layer of our models to pretraining the entire model with hierarchical representations. If learning word vectors *{e.g. Word2vec}* is like only learning edges, these approaches *{e.g. ELMo}* are like learning the full hierarchy of features, from edges to shapes to high-level semantic concepts." In essence these new techniques have a much richer semantic representation of words/sentences and thus enable the labeling of text with a more human-like nuanced understanding of words.

Possibly more exciting is the idea that with these newer systems we may be able to build transferable pre-trained universal word/sentence embeddings that we can use for virtually any model and achieve much better accuracy. (This sounds a lot closer to a human understanding of text to me!)

These newer systems are poised to replace Word2vec and other currently popular word embeddings in a short period of time. But only time will tell how well they capture meaning.

So what are full-text AI model builders and Data Scientists to do?

Organizations know that they need labeled data to produce an accurate model. But the NLP community lags far behind the computer vision community in terms of the availability of fully labeled training datasets (e.g. ImageNet). In addition, there is no consensus as to the best way to label text for use in AI models. This means you need to work with an experienced partner that can help you decide what techniques are optimal for your unique situation. Creating an automated data labeling process for your specific model & domain will take time and effort to build. But this effort is minimal compared to the cost of using people to create a labeled training dataset, or the cost of using the wrong techniques to produce a poor quality model. Also remember, an automated process can label new data in real time (e.g. streaming), which is critical in the use of many AI models. Given the complexities of NLP models, we recommend working with an organization with lots of experience. At Informatics4AI, we are using tools that process billions of text documents each and every day and have processed over a trillion documents overall. We understand text, we understand how to help you use advanced NLP techniques for predictive lift, and we have the experience to ensure your full-text AI project succeeds. Give us a call, we're here to help.

About US

Informatics4AI helps organizations improve the performance of AI models built on textual datasets (such as chatbots and conversational systems) using automated ontology generation and entity level sentiment detection. We transform raw text into labeled datasets that a machine can understand with as much accuracy as a human. Informatics4AI has over 30 years of experience in managing text, we are using tools that process billions of text documents each and every day and have processed over a trillion documents overall. We use these same techniques to improve search systems. Give us a call, we're here to help your AI project and search system succeed.

Informatics4AI
PO Box 380054
Cambridge, MA 02238
(617) 453-8220