Ontology Driven Machine learning Approach for Disease Name Extraction from Twitter Messages

Mark Abraham Magumba*, Peter Nabende† and Earnest Mwebaze§
Department of Information Systems,
College of Computing and Information Sciences, Makerere University,
P. O. Box 7062, Kampala, Uganda
e-mail: *magumbamark@hotmail.com, † peter.nabende@gmail.com, §emwebaze@cit.ac.ug

***Abstract*—Twitter and social media as a whole has great potential as a source of disease surveillance data however the general messiness of tweets presents several challenges for standard information extraction methods. Current methods for disease surveillance on twitter rely on inflexible keyword based approaches that require messages to be pre-filtered on the basis of a disease name which is supplied a priori and are not capable of detecting new ailments. In this paper we present an ontology based machine learning approach to extract disease names and expressions describing ailments from tweets which may be employed as part of a larger general purpose system for automated disease incidence monitoring. We also propose a simple methodology for automatic detection and correction of errors.**

Keywords-named entity recognition; knowledge engineering; ontology; epidemiology

I. INTRODUCTION

Current methods of disease surveillance on twitter rely on keyword filters that limit them to particular diseases [1-3] rather than automatically recognizing arbitrary strings as candidate disease names. In Natural language processing parlance this problem falls under the task of named entity recognition. State of the art machine learning approaches regularly obtain performance in excess of 80% accuracy on gold standard hand annotated datasets on this particular task. To achieve these results they heavily rely on word based features such as the words themselves and their contexts, whether or not words are capitalized, the distance of givens word from definite articles, suffixes, prefixes [4-6], word embeddings [7] and so on. These features are predicated on the condition that the training and testing data have a similar lexicon and writing style. On a medium like twitter such an approach is impractical as each tweet could potentially be authored by a different individual meaning that even in a small dataset there are bound to be large variations in writing style between different users. Furthermore, twitter text has a large lexical diversity which is exacerbated by the fact that words are often misspelled and there is frequent use of slang [8].

In our approach we employ two key intuitions; firstly that for the purpose of communicating diseases some words are more important than others. Secondly, that words themselves are representative of some higher mental concepts and that in the bounds of a given topic of discourse there is a fairly small number of concepts but a possibly infinite number of words and ways to express them. Combining these two intuitions, we theorize that people tend to use similar words to communicate illness and consequently attempt to construct a dictionary of disease. We then organize these words into concepts in an ontology that we employ for semantic annotation of tweets.

Figure 1 below is a visualization of the word frequency in four data sets containing tweets that mention a selection of diseases. The frequency of words is encoded as their font size and this visualization shows the 100 most frequent words per data set. In producing this visualization we also removed certain elements like URLs and stop words in addition to the names of the actual diseases.

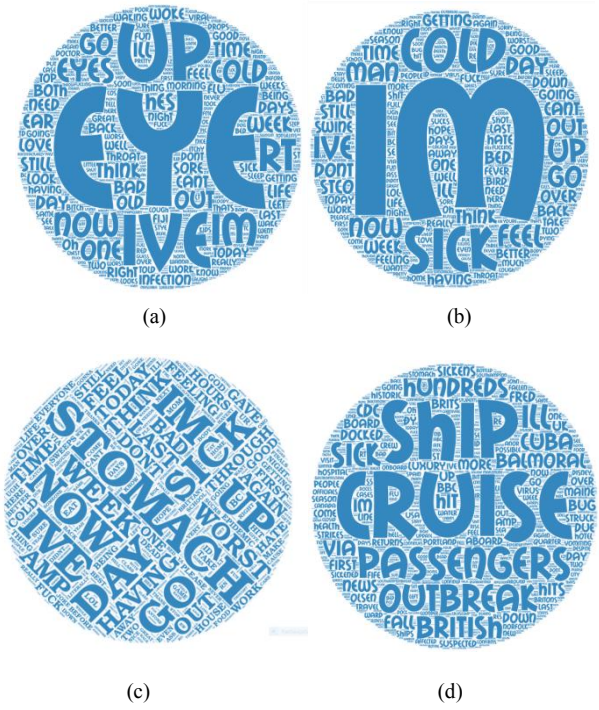

Figure 1. Word clouds generated from data sets

From the preceding figure it is quite clear that these words can easily be grouped into categories. For instance "eye" is prominent in the conjunctivitis dataset (a) as is "stomach" in gastroenteritis dataset (c) and with some domain knowledge we can easily infer that these symbolize the sites of infection for the corresponding diseases. With this knowledge we can create a category for anatomical references. Furthermore, even though we do not encounter

978-1-5386-2030-4/17/$31.00 ©2017 IEEE

the word "leg", as humans we can easily predict that it too is a member of this category and hence extend it.

However, even though the term "cell nucleus" refers to a part of the body it is way too technical and specific to expect it to occur in casual twitter posts with significant regularity therefore it is our opinion that we can safely ignore it and simply register it as a noun. In deriving concepts for the ontology we repeat this process several times until we obtain a set of concepts that we think are representative of the domain.

A. Related Work

Previous attempts to detect disease names in natural language text have been made on less noisy technically oriented sources such as discharge summaries, echocardiogram, radiology, and ECG reports [9] and specialized databases such as DistiLD [10] and COSMIC (Catalog of Somatic Mutations in Cancer)[11] as described in [12]. The techniques range from machine learning methods like Conditional Random Fields [9] to dictionary lookup techniques [12]. In these cases because the text originates from predominantly technical writers semantic annotation is fairly straight forward using any of the several publicly available medical thesauruses like UML (Unified Medical language) [9].

We are currently not aware of any work that has attempted this task on tweets. A method developed by Micheal Paul and Mark Dredze [13] is able to partition a corpus of tweets talking about different ailments into clusters that contain tweets that roughly describe the same ailment using a variation of Latent Dirichlet Allocation however it doesn't directly return the actual ailment being talked about. Our objective here is to develop a model that given such a corpus can enumerate the ailments described in it.

II. Materials and Methods

We employ two related sequence modelling techniques namely Conditional Random fields (CRFs) and a log linear model.

A. Conditional Random Fields

CRFs are by far the most popular machine learning technique for Named Entity although work has also been done using log linear models [14, 15] so we included them here for comparison. A Conditional Random field follows from a General Gibbs distribution, which is a form of undirected graph, conditioned on the outcomes to be predicted. That is given some input variables {X1…..Xn} and a set of target variables {Y1….Yn}, where the predictor variables are assumed to be heavily correlated the following equations describe CRFS:

$$\varphi = \{\varphi_1 D_1 \, , \varphi_1 D_1 \; \ldots\ldots \varphi_n D_n \} \quad (1)$$

$$\tilde{P}_{\varphi}(X,Y) = \prod_{i=1}^{\varphi_n D_n} \quad (2)$$

$$Z_{\varphi}(X) = \sum_{Y} \tilde{P}_{\varphi}(X,Y) \quad (3)$$

$$\tilde{P}_{\varphi}(Y|X) = \frac{1}{Z_{\varphi}(X)} \tilde{P}_{\varphi}(X,Y) \quad (4)$$

Equation 2 describes the general Gibb's distribution given the set of factors in equation 1, equation 3 describes a partition function for a given input variable X and equation 4 describes the distribution for a given value of the target variable Y given X. This conditional distribution is normalized by the partition function. The CRF is a family of such conditional distributions

B. Log Linear Model

The log linear model defines the corpus in terms of features where features are functions that map a given set of inputs and outcomes to some real value, usually features are binary where x is some sequence of contextual information such as the labels for words $i - k$ to the k^{th} term inclusive and y is the k^{th} term and i is the size of the context window. The features are usually binary valued that is if a sequence x is followed by y then $f(x, y) = 1$ *and* 0 otherwise. Therefore, we have some input domain X and a set of labels Y conditioned on X, the aim is to estimate the conditional probability $p(y|x)$ for any $x \in X$ and $y \in Y$. The features are stored in a feature vector; in addition parameters are defined for each feature and stored in a corresponding parameter vector V. The log linear model expresses the probability of a given label or outcome y given some input sequence x and a given value of v as follows:

$$p(y|x;v) = \frac{e^{v.f(x,y)}}{\sum_{y' \in Y} e^{v.f(x,y')}} \quad (5)$$

The denominator is a normalization term which ensures that the value is always between 0 and 1 and for all y the sum is 1. This can be rewritten in logarithmic form as follows:

$$\log p(y|x;v) = v.f(x,y) - log \sum_{y' \in Y} e^{v.f(x,y')} \quad (6)$$

C. An Ontology of Disease Communicating Words

We model a sub set of language comprising the important semantic concepts required to communicate illness as conceptual objects. These concepts are of two broad categories, those that directly describe disease incidence such as references to disease causing organisms such as bacteria and general linguistic terminology like words that imply negation and references to temporality such as words like "now" and "then" and descriptions of space and time. As depicted in figure 2, at the top of the hierarchy everything is

conceptualized as an object, there are two types of objects that is, real objects and abstract objects. Real objects refer to tangible things and abstract objects refer to concepts like time, negation and events.

Living things comprise the key biological actors such as hosts (the organism that suffers disease which in this case is a person), pathogens (disease causing organisms) and vectors (disease spreading organisms). Inanimate objects are of two major classes namely environments and substances. We deliberately exclude technical terminology such as names of specific diseases or drugs and purely concentrate on the basic linguistics of communicating diseases.

There are three types of concepts namely relationships and properties and actions. Relationships describe interactions between concept classes in the object hierarchy and correspond to OWL (Web Ontology Language) object properties whereas properties describe object and relationship attributes and correspond to OWL data properties. Actions describe object behavior for certain objects like people where a typical action is to exercise and can be thought of as data properties for which the range and domain are the same.

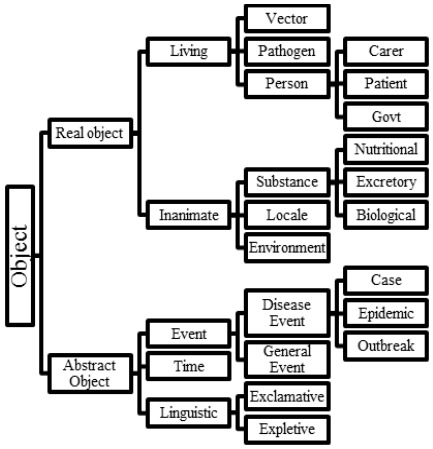

Figure 2. Partial Class hierarchy for Human Disease language Ontology

1) Concept Dictionaries

Finally, for each conceptual object be it a relationship, object or action there is a corresponding concept dictionary. In figure 3 below depicts a partial decomposition for the people conceptual object with corresponding concept dictionaries indicated in curly brackets. The full ontology along with a description is publically available in OWL/XML format at our Github repository[1]. The concept dictionary is simply a list of words related to the concept. Roughly speaking the members of a given concept dictionary are related by three relationships namely synonymy, hyponymy/hypernymy and meronymy. Synonymy refers to semantically equivalent words such as "skin" and "dermis", hypernymy is refers to increasing generalization increasing generalization for instance "location" is a hypernym of "house", hyponymy is the inverse of hypernymy as in "house" would be a hyponym of "location".

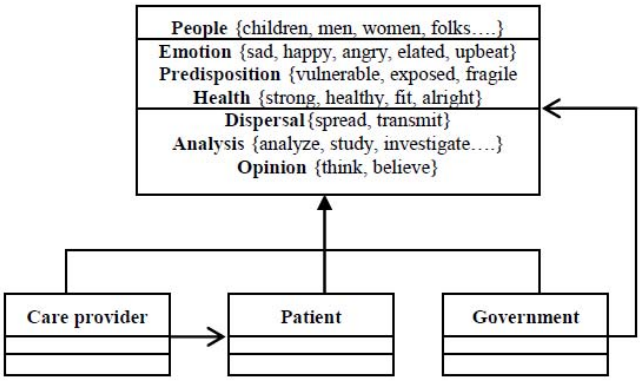

Figure 3. Partial Decomposition of People Concept Sub Hierarchy with partial dictionaries, the concept dictionaries are indicated as lists of related words shown here in curly brackets.

Generally speaking immediate hyponyms and hypernyms are grouped together with concept synonyms. Meronyms are similarly grouped together. Meronymy implies a part of relationship for instance "arm" is a meronym of "body".

In a few cases we have applied some subjective judgments like expectations on the expected frequency of words belonging to a given concept to bundle words on some other thematic criteria. As an example there is a treatment concept which bundles together any treatment related word such as treatment nouns like stethoscope, syringe and medicine and treatment verbs like treat, therapy. The result is a fairly coarse grained classification.

III. THE DISEASE NAME EXTRACTION PIPELINE

Next we describe the steps taken to extract disease names from a corpus of tweets, figure 4 below summarizes the procedure. A disease name could be a single word like pneumonia or a multi word expression like the winter vomiting bug or man flu.

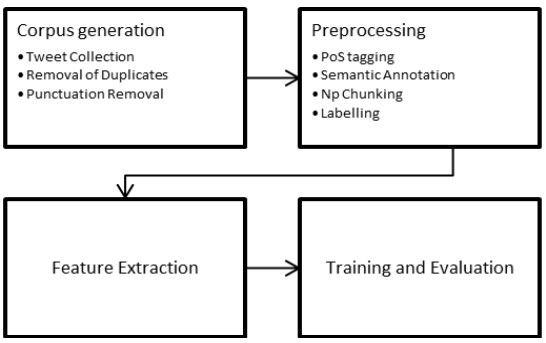

Figure 4. Disease Name Recognition Pipeline

[1] https://github.com/MarkMagumba/Twitter-Disease-incidence-Description-Language-Ontology

A. Corpus Generation

The first step is the creation of the corpus. We obtain tweets from a basic twitter account using some specific keywords via a python script through Twitter's Streaming API [16] using python's tweepy plugin. The tweets we download are those that are marked as public, this is the default security level and they are only marked private if expressly indicated by users. We employ simple keyword filters to extract the desired tweets. These keywords are names of diseases like flu and pink eye.

Not all messages mentioning diseases necessarily talk about them in the context of reporting active cases of disease incidence and since this is the context we are interested in, we see no point in training our model on tweets that talk about diseases in these alternative contexts. Therefore, we further examine the tweets manually to extract only those that report ongoing or recent cases of disease activity within some time window of up to a few weeks where we assume that diseases may still be in their communicable period. Using this approach we generate three datasets: An influenza + common cold + Listeria data set which comprises 73, 848 tokens in 7835 tweets which we employ as the training data set, a mumps + measles + pneumonia data set which comprises 3,866 tokens in 316 tweets a pink eye data set which comprises 3,428 tokens in 267 tweets.

The mumps + measles + pneumonia and the pink eye data sets are used for testing. The former is used to test the models on cases where the disease name is a singular word and the latter to assess the performance where the disease name is a multi-word expression. In addition we eliminate duplicates by removing retweets, (tweets with the "RT" tag) and also manually checking for duplicates that may not be marked as "RT" and finally we remove all punctuation except the "#" and "@" symbols where they appear at the beginning of tokens where they are used to denote hashgtags and users respectively.

B. Pre-processing:

The next step is pre-processing. We start off by Part of Speech tagging the tweets. We employ the GATE (General Architecture for Text Engineering) twitter tagger [17]. The tagger uses the Penn Treebank tag set [18] in addition to three additional tags "HT", "USR" and "URL" corresponding to twitter specific phenomenon namely hashtags, users and URLs. The next step is semantic annotation, for this we employ our ontology of concepts depicted in figure 2. Each concept has a corresponding concept dictionary which comprises semantically related words related to the given concept.

For each word in each tweet we obtain the semantic category simply by looking up the concept that contains it in the ontology. Our ontology comprises 1531 words corresponding to 136 concepts. Needless to say compared to the possible number of words that may occur in English this is a small number of words and most words will not appear in the ontology. We mark these words as "OOV" or out of vocabulary.

This is followed by Noun Phrase Chunking (NP-Chunking). Most entity mentions if not all will appear in the Noun-phrase potion of a sentence therefore it helps the accuracy of our model if we indicate the NP chunks for it. For this a simple regular expression chunker is sufficient. We label any sequence of numbers, conjunctives, adjectives followed by any number of nouns as a noun phrase ("NP") and anything else is simply labeled "O" for out.

The final step is labeling, we give each word a label of ET (Entity True), PERS for persons, ORG for organizations, PLACE for geo-political entities and MC (Miscellaneous category) for everything else. The Entity True labels refer to words that are part of a disease name. Table 1 shows the final annotation for the tweet, "83 likely cases of mumps in Washington"

TABLE I. EXAMPLE ANNOTATION OF A TWEET

Word	*PoS tag*	*Semantic Category*	*NP-Chunk*	*Label*
83	CD	OOV	NP	MC
likely	JJ	PROBABILITY	NP	MC
cases	NNS	INCIDENCE	NP	MC
of	IN	OF	O	MC
mumps	NN	OOV	NP	ET
in	IN	VIN	O	MC
Washington	NNP	OOV	NP	PLACE

C. Feature Extraction and Training:

For training the CRF we use only the part of Speech tag, Semantic category and NP-chunk using both unigram and n-gram features. We use a large context window of 6 for the history as well as look ahead features. For training the log linear model we use a smaller window using only the part of speech tag and semantic category of the previous three words of history and those of the next two words for each word. The reason for this is that the log linear model implementation suffered more performance wise in terms of memory utilization as the number of features was increased and became infeasible beyond this number of features.

The final models disregard the actual words as these are replaced by the semantic category of each word as per the corresponding ontology concept. Each word maps to some concept or the "OOV" tag, effectively we have a fixed length lexicon as any tweet can be represented in terms of our ontology concepts thereby eliminating the effects of lexical divergence between the learned models and new data. For the log linear model we used the Stanford part of speech tagger [19] and for the CRF we employed Naoaki Okazaki's CRF++ package [20].

IV. RESULTS AND DISCUSSION:

The results in table 2 below depict the tag wise performance with respect to the ET tag, the CRF generally

outperforms the log linear model although as explained in the previous section we are able to use more features with the CRF software because it is computationally inexpensive. For both approaches precision is significantly higher than recall for both data sets

TABLE II. PERFORMANCE OF MODELS ON TEST DATA SETS

Algorithm	***Metric***	***Data set***	
		Pink eye	Measles + pneumonia + mumps
Log linear model	Precision	0.71	0.68
	Recall	0.58	0.43
	F1 score	0.64	0.53
Conditional random fields	Precision	0.68	0.79
	Recall	0.63	0.58
	F1 score	0.65	0.67

V. ERROR ANALYSIS:

At first glance the results do not look remarkable; however a closer analysis of the errors reveals an interesting distribution. As is clear in figure 5, the errors are extremely spread out that is the models do not consistently mislabel any given word as a disease when it is not and are far more likely to label a word which is a disease with more consistency.

Those words that are correctly labeled as diseases stand out as spikes on the plot. The implication is that a simple thresh-holding approach to error detection and correction is feasible. For instance if we set a threshold of 5, that is if we set the system to disregard any words that are assigned the ET tag 5 times or less we effectively eliminate all false positives. Furthermore, some of the errors are false errors meaning the model is actually correct but the tokens were mislabeled during annotation. These are mainly of two types.

The first category is misspelled tokens which we missed since we heavily relied on semi-automatic approaches to make the annotation task more manageable as opposed to gold standard hand annotated data sets, the second category include cases where the disease name is inside a hashtag like #measles because we generally ignored hastags as we did not have an efficient method to properly normalize them particularly in situations where you have mult-word hastags such as #measlesoutbreak. Therefore, given better text normalization techniques it is more than likely that we would have better performance.

Another source of error is poor part of speech tagging. Since the procedure is heavily reliant on having the correct part of speech tags as this directly affects the NP-chunking task which directly affects the final entity extraction. For state of the art performance gold standard data sets rely on hand annotation but we had no such luxury and mostly relied on the tagger described in [10] but still found several errors on inspection such as verbs labeled as nouns.

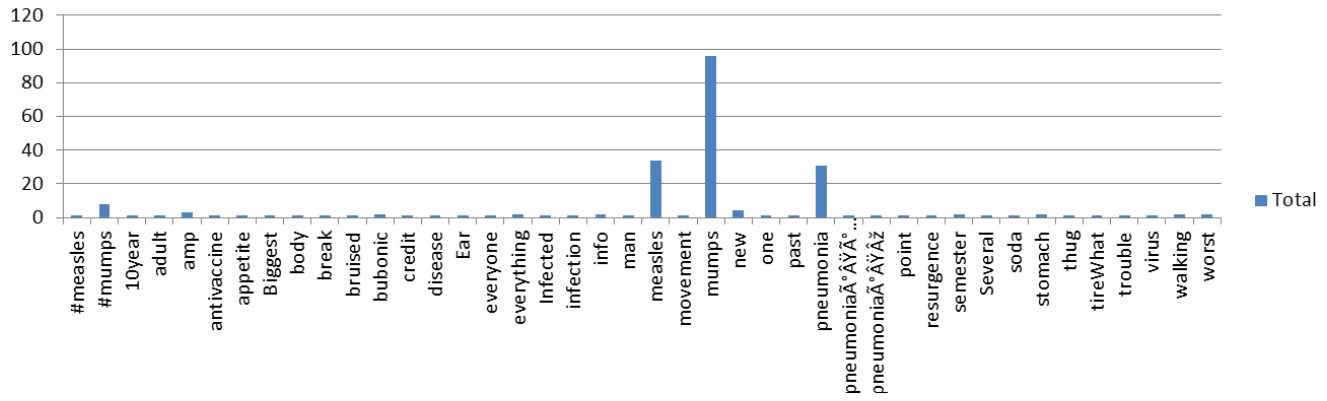

Figure 5. Frequency distribution of words tagged as diseases for the measles + mumps + pneumonia dataset, the y axis indicates the number of times a word is tagged as a disease, the x axis indicates all those words that are tagged as diseases

Furthermore, we were unable to use the best model supplied because it was prohibitively slow. The alternative optimized model however did not define any closed categories[2] meaning any tag could be assigned to any word and indeed there were cases where random nouns were tagged as pronouns. We corrected as many of these cases as we could but by no means do we expect to have eliminated all errors of this sort. Nonetheless, we do not believe it is particularly important that the system obtains extremely high performance but rather that its performance behavior is such that errors can easily be detected as described above.

VI. CONCLUSION

These experiments were meant to provide proof of concept for our proposed approach and therefore the results should not be taken as a performance upper limit. We

[2] A closed class is a grammatical category whose membership is closed for instance Definite articles which comprises a single word – the

believe better performance is obtainable by further refining the ontology and other improvements like better text normalization and handling of hashtags. This approach is particularly useful on a medium like twitter where it may not be possible to exhaustively enumerate all the different ways individuals may express disease conditions for the discovery of previously un-encountered expressions for instance newly emerging diseases.

For known expressions a simple word lookup would be both more efficient and more accurate especially if robust text normalization is achievable. The ideal system would employ a two-step approach by first attempting a word lookup against suitably filtered messages and then applying the ontology driven machine learning approach described here if the first approach does not return a hit.

REFERENCES

[1] K. Lee, A. Agrawal and A. Choudary, "Real Time Disease Surveillance Using Twitter Data: Case Study Flu and Cancer", in proceedings of the 19th ACM SIGKDD conference of Knowledge discovery and Data Mining. August 11-14, 2013, Chicago, Illinois, USA , ACM. 2013 , pp. 1474 – 1477

[2] Google Inc, [online] Available https://www.google.org/flutrends/about/ [Accessed 25-05-2017]

[3] V. Lampos and N. Cristianini, 'Tracking The Flu Pandemic By Monitoring The Social Web'. In proceedings of 2nd Workshop on Cognitive Information Processing (CIP 2010), June 14-16, 2010. Naregno Elba island Italy, IEEE [online] Available http://ieeexplore.ieee.org/abstract/document/5604088/ [Accessed 25-05-2017]

[4] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. "Named entity recognition through classifier combination". In Proceedings of the seventh conference on Natural language learning at HLT-NAACL, May 31st- June 1st, 2003, Edmonton, Canada,. ACL, 2003. pp. 168–171

[5] H. L. Chieu. Named entity recognition with a maximum entropy approach. In Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003), 2003, Edmonton, Canada, ACL pp. 160–163

[6] R. K. Ando and T. Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data". JMLR, vol. 6. pp. 1817 – 1953, November, 2005 [online] Available http://www.jmlr.org/papers/volume6/ando05a/ando05a.pdf [Accessed 25-05-2017]

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa. "Natural language Processing (almost) from scratch", JMLR, vol. 12, pp. 2493-2537 [online] Available http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf [Accessed 25-05-2017]

[8] J. Eisenstein, "What To Do About Bad Language on the Internet", In Proceedings of NAACL-HLT, June 09-14, 2013, Atlanta, Georgia, pp: 359–369

[9] O. Ghiasvand, "Disease Name Extraction in Clinical Text Using Conditional Random Fields" Msc. [Dissertation] University of Wisconsin-Milwaukee, USA. [Online] Available http://dc.uwm.edu/etd/495/ [Accessed 25-05-2017]

[10] A. Pallejà, H. Horn, S. Eliasson and L.J. Jensen, "DistiLD Database: Diseases and Traits in Linkage Disequilibrium Blocks" Nucleic Acid research, vol. 40, pp. 1036 –1040. January 2012

[11] S.A. Forbes, G. Bhamra, S. Bamford, E. Dawson, C. Kok, J. Clements et al. "The Catalogue of Somatic Mutations in Cancer (COSMIC)" in Current Protocols in Human Genetics, John Wiley & Sons, 2008

[12] S. Pletscher-Frankilda, A. Pallejàa, K. Tsafoua, J.X. Binder and J. Jensen, "Diseases: Text Mining And Data Integration Of Disease–Gene Associations" In: Methods, vol. 74, M. Andrade-Navarro and C. Perez-Iratxeta. Eds., Elsevier, 2014. pp. 83-89.

[13] M.J. Paul and M. Dredze, "Discovering Health Topics in Social Media Using Topic Models", PLoS ONE , vol. 9, no. 8. August, 2014. [online]. Available http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0103408 [Accessed 25-05-2017]

[14] H.L. Chieu, "Named entity recognition: a maximum entropy approach using global information". In Proceedings of the 19th international conference on Computational linguistics - Volume 1 (COLING '02), August 24th – September 1st, 2002, Taipei, Taiwan. ACL, Stroudsburg, PA, USA, pp. 1-7

[15] O. Bender, F.J. Och, and H. Ney, "Maximum Entropy Models for named Entity Recognition" In Proceedings of the 7th Conference on Natural Language Learning, May 31st – June 1st, 2003, Edmonton, Canada. ACL, Stroudsburg, PA, USA. Pp. 148-151

[16] Brightplanet.com, "Twitter Firehose Vs Twitter API: What's the Difference and Why You Should Care?", June 25th 2013 [online] Available https://brightplanet.com/2013/06/twitter-firehose-vs-twitter-api-whats-the-difference-and-why-should-you-care/ [Accessed 25-05-2017]

[17] H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan, "Gate: An Architecture For Development Of Robust HLT Applications", July 7th – 12th, 2002. Philadelphia, USA, ACL, Stroudsburg, PA, USA, pp. 168 - 175

[18] A. Taylor, M. Marcus and B. Santorini, 'The Penn Treebank: An Overview', in Treebanks: Building and Using Parsed Corpora, A. Abeille, Ed. Netherlands, Dordrecht. Springer. 2003. pp: 5-22

[19] K. Toutanova and C.D. Manning, "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger" in Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics – volume 13, October 7th – 8th, 2000. Hong Kong, ACL, 2000, ACL, Stroudsburg, PA, USA, pp. 63-70

[20] N. Okazaki, "CRFsuite, A fast implementation of Conditional Random Fields (CRFs)", 2007. [online]. Available http://www.chokkan.org/software/crfsuite/ [Accessed 25-05-2017]