Automated Learning of Everyday Patients' Language for Medical Blogs Analytics

Giovanni Stilo
Dipartimento di Informatica
stilo@di.uniroma1.it

Moreno De Vincenzi
Dipartimento di Informatica
devincenzi@di.uniroma1.it

Alberto E. Tozzi
Ospedale Pediatrico Bambin Gesù
albertoeugeniotozzi@opbg.net

Paola Velardi
Dipartimento di Informatica
velardi@di.uniroma1.it

Abstract

Analyzing how people discuss about health-related topics on dedicated forums and social networks such as Twitter, can provide valuable insight for syndromic surveillance and to predict disease outbreaks. In this paper we present a minimally trained algorithm to learn associations between technical and everyday language terms, based on pattern generalization and complete linkage clustering, and we then assess its utility on a case study of five common syndromes for surveillance purposes.

1. Introduction

Infodemiology is defined as "the science of distribution and determinants of information in an electronic medium, specifically the Internet, with the ultimate aim to inform public health and public policy" (Eysenbach, 2006). A seminal work in this area is (Ginsberg et al., 2009), in which the level of influenza in the U.S. is estimated using the relative frequency of search queries related to influenza-like illness. Similarly, in (Althouse et al., 2011), the authors demonstrate that query search volumes associated to Dengue fever can predict the incidence of Dengue. Another recent study (Xu et al., 2011) analyses the problem of predicting the tendency of hand-foot-and-mouth disease (HFMD), clustering HFMD-related search queries, medical pages and news reports. Query search volumes are estimated using Google Trends (GT) [1] or Google Flu, however, forums and micro-blogs (like Twitter) appear to be a better source of information, since keywords occur in contexts. Contexts make it possible to use text mining techniques for sense disambiguation, topic filtering and mood analysis (Berendt, 2011; Corley, 2009; Von Etter et al., 2010; Cohen and Hersh, 2005; Paul and Dredze, 2011). Among the others, the problem of tracing patient's naïve medical terminology is a very crucial one (Dahm, 2011; Molina Healthcare, 2004). Consider the following striking difference in the usage of terms describing the same health conditions, the first by a clinician, the second by a patient: "*Clinicians should maintain a high index of suspicion for this diagnosis in patients presenting with influenza-like symptoms that progress quickly to respiratory distress and extensive pulmonary involvement.*"[2] "*For the past 3 days I have had a stuffy, runny nose, congested chest, fever, sore ears and throat and burning eyes. I've been taking cold and flu medication, and it doesn't help*"[3]. Clearly, the patient's symptoms should induce "*a high index of suspicion*", but for an automated system to capture a similarity between the two symptom descriptions is not obvious. Being able to understand the way people talk about their health conditions in "peer to peer" communications is crucial for an effective monitoring of health-related behaviors based on social data.

In this paper we present a minimally supervised algorithm to learn patient's jargon and we apply it to the analysis of 5 common syndromes. We obtain an impressive correlation with existing official data, and furthermore, we are able to monitor not only a disease outbreak, but its related symptoms, which is a clear advancement over previous works in this area. The paper is organized as follows: in Section 2 we present the algorithm in detail, in Section 3 we describe the corpora and tools used to monitor patients' discussions and we analyze five cases of interest for epidemiologic surveillance. Section 4 is dedicated to the

[1] http://www.google.com/trends/

[2] *www.ncbi.nlm.nih.gov/pubmed/20085663*

[3] *ehealthforum.com*

Proceedings of Recent Advances in Natural Language Processing, pages 640–648,
Hissar, Bulgaria, 7-13 September 2013.

analysis of related work, and Section 5 presents our concluding remarks.

2. Mapping Medical Jargon And Everyday Language

In this Section we present a minimally supervised algorithm to learn from the web (Wikipedia, Google snippets, and other resources) a set of generalized patterns to establish a correspondence between technical and naïve jargon, and to identify common expressions used by patients to describe their medical conditions. The algorithm starts with a relatively small learning set *MC* of medical conditions, composed by pairs (tt_i, nt_j), where tt_i is a technical term and nt_j a naïve term[4], e.g. <*myocardial infarction, heart attack*>, <*emesis, vomiting*> etc. The set *MC* is divided in three subsets S_o, S_1 and S_2 used for learning, refining and testing. The algorithm has four steps:

1. **Web mining step**: using S_0, we extract from the Web sentence snippets including both terms;
2. **Clustering step**: we generalize lexical patterns between a tt_i and an nt_j (or vice versa) and create weighted clusters of similar patterns; we also learn generalized expressions for tt_i and nt_j;
3. **Reinforcement step**: using S_1, we test the precision and recall of each pattern and adjust cluster weights;
4. **Testing phase**: The algorithm is tested on S_2 and the steps are repeated for any possible permutation of S_o, S_1 and S_2.

As a preliminary step, we define a policy to generalize lexical patterns and terminological expressions for medical conditions, as well as a distance measure to compute the similarity between patterns. Let tt_i and nt_j be single or multi-word expressions describing a technical or naïve medical condition, respectively, and let $p = w_1, w_2, ...w_{|p|}$ be a word sequence between them, found on some document or web resource, e.g. "*abdominal obesity, colloquially known as belly fat*". Note that we can have $tt_i < p > nt_j$, as in previous example, or $nt_j < p' > tt_i$ as in "*belly fat is known clinically as abdominal obesity*". A pattern p is generalized as $p' = w_1', w_2', ...w_{|p|}'$ where:

$$(1)\quad w_i' = \begin{cases} w_i^* \text{ if } POS(w_i) \in \{NOUN, VERB, PREP, PUNCT, \text{"or"}\} \\ POS(w_i) \text{ otherwise} \end{cases}$$

where w_i^* is the word lemma and $POS(w_i)$ is the part of speech obtained with a POS tagger[5]. For example, if p="*is another word for*", then *p'=be #DT word for*. Since tt_i and nt_j are often multi-word expressions, e.g. "*high level of potassium*", we apply pattern generalization also to these terminological strings. A multi-word expression for a term describing a medical condition is generalized as follows:

$$(2)\quad w_i' = \begin{cases} BODYPART \text{ if } w_i \in \{eye, nose, skeleton..\} \\ DISCOMFORT \text{ if } w_i \in \{pain, itch, ache, miserable, ...\} \\ else\ w_i' = w_i \text{ if } freq(w_i) > \vartheta \\ else\ w_i' = POS(w_i) \end{cases}$$

For example, *muscle weakness, heart attack, hair fungus*, generalize as BODYPART #NN. Discomfort words and body parts have been retrieved from publicly available Web resources[6]. The third generalization rule in (2) captures additional frequent words such as *illness, inflammation, infection, etc*. Rules in (2) are used to learn generalized sequences s_k for medical conditions, using the examples in *MC*, and group them by frequency. We denote with *T* the set of learned generalized medical condition patterns. Table 1 shows some of the most frequent sequences.

Sequence	Examples
NN	bilharzia, fainting, clenching, chickenpox
BODYPART NN	muscle weakness, heart attack, hair fungus
JJ BODYPART	crooked tooth, stuffy nose, crooked back, dry mouth
inflammation of BODYPART	inflammation of the heart, inflammation of the liver, inflammation of the skin

Table 1. Four most frequent generalized sequences for medical conditions (both *tt* and *nt*)

[4] In what follows, whenever a preposition applies to either a technical term or a naïve term, we use the notation *t* and *pt* (term and partner term) or *ct* (candidate partner term).

[5] We use the Treetagger *http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/*

[6] E.g. for discomfort: http://www.macmillandictionary.com/thesaurus-category/british/Physically-painful-and-describing-pain

Given a pattern *p*, we define three categories for its elements *w*:

- $A := \{w_i \in p \mid POS(w_i) \in \{NOUN, \{VERB \neq be, can..\}\}\}$
- $B := \{w_i \in p \mid POS(w_i) \in \{PREP, ADJ, PUNCT\}\}$
- $C := \{w_i \in p \wedge w_i \notin \{A, B\}\}$

Let w^A, w^B and w^C be three experimentally tuned weights assigned to the word categories A, B and C. Given two patterns p_i and p_j, the *distance* between the patterns is defined as:

$$(3)\ d(p_i, p_j) = 1 - (count(p_i, p_j, A) \times w^A + (count(p_i, p_j, B) \times w^B + (count(p_i, p_j, C) \times w^C)$$

where $count(p_i, p_j, A)$ is the amount of common words in the two patterns belonging to category A. Matches in category A have a higher relevance wrt those in the other categories. For example, if the weights are 0.55, 0.3 and 0.15 respectively, *d*("*, known in medical terms as*", "*is another term for*")=0.725 and *d*("*, medical term for*", "*is fancy term for*")=0.25.

Learning Clusters Of Patterns

During step 1 of the algorithm (*web mining*), we start with S_o, and we extract from the Web text snippets including the pairs in S_o. Then, we take the *word sequence* between the two terms, and we apply pattern generalization using the rules in (1). To reduce noise, we also discard sequences whose length is more than 7 tokens, an experimentally selected threshold. Let *P* be the set of survived different patterns. For each pattern $p_i \in P$ we compute a score corresponding to the normalized count of different seed pairs that supported the pattern, e.g.:

$$(4)\ weight(p_i) = \frac{|distinct\ seed\ pair\ with\ p_i|}{\max_j(|distinct\ seed\ pair\ with\ p_j|)}$$

Next, we apply *pattern clustering* (step 2). For pattern clustering, we use an approach called *complete linkage* (Jain, 2010). The clustering literature is immense, and many other algorithms are available: however, complete linkage avoids the so-called *chaining phenomenon*, which causes one cluster to attract most of the population members. Furthermore, unlike the majority of clustering algorithms, complete linkage is not heavily parametric[7]. In complete linkage, the similarity of two clusters is defined as the similarity of the most dissimilar members, which is equivalent to choosing the cluster pairs whose merge has the smallest diameter. The algorithm starts with singleton clusters (e.g. each composed by one pattern $p \in P$) and then progressively merge two clusters C_i and Cj into larger ones, according to the distance function: $D(C_i, C_j) = \max_{p_i \in C_i, p_j \in C_j} d(p_i, p_j)$, where $d(p_i, p_j)$ is defined in our case by the formula (3). Using complete linkage we obtain balanced clusters, with low dissimilarity among the members of each cluster, for example: "*is a very broad term defining" "is a general medical term used for" "is a general term for" "is the common term for*", etc. Conversely, very specific patterns (e.g. "*your doctor would call it*") have the tendency to remain isolated. We define the following measure to weight the quality of the derived clusters: $score(C_i) = \sum_{p_j \in C_i} weight(p_j)$ where $weight(p_j)$ is computed as in formula (4).

Term Matching And Cluster Refinement

Term matching is the process of finding one or more candidate partner terms *ct* for a term *t*, where *ct* is technical if *t* is naïve, or vice versa. Once a clustering $C: \{C_1, C_2..C_k\}$ has been learned, it is used to find unknown technical or naïve terms in the following way: we take a term *t*, for example *belly fat*, and seek in the web for *domain relevant* sentences with this term. As a preprocessing step, we eliminate sentences not in the medical domain (e.g. if t=*plague*: "*The capacitor plague (also known as bad capacitors or "bad caps") was a problem with a large number of premature failures of aluminum electrolytic capacitors ...*") using a domain heuristics. The formula is the following: $DomainWeight(s) = \frac{|B(s) \cap D|}{|B(s)|}$ where *B(s)* is the bag of words of the retrieved snippet, and *D* is a set of singleton words (only nouns) extracted from a medical terminology[8]. Sentences with a domain weight lower than a threshold α are discarded. We then identify to the left or to the right of *t* the candidates partner terms *ct*. For example, given the sentence (retrieved for *t=belly fat*): "*abdominal obesity , colloquially known as* ***belly***

[7] For example, in many algorithms the number of clusters k is a parameter.

[8] We use Freebase http://www.freebase.com/view/medicine/disease, but any other medical terminology can be used.

fat *or central obesity*" two candidates are selected, *abdominal obesity* and *central obesity*. To select candidates the algorithm uses a chunker[9] to identify noun phrases, and then select the best matching NP in terms of likeliness, using the set *T* of generalized learned sequences for medical conditions (see Table 1). This allows e.g. to prefer *central obesity* rather than *obesity* alone. For each candidate partner *ct* of *t (*e.g. *abdominal obesity)*, we take the pattern *p* between *t* and *ct* (*", colloquially known as"*), and compute its distance wrt the previously acquired cluster members, according to:

$d'(p,C_j) = \frac{\sum_{p_i \in C_j} d(p,p_j)}{|C_j|}$ the most similar cluster is

therefore: $C_p^* = \begin{cases} C_k \, if \;\; p \in C_k \\ arg\min_{C_j \forall j} d'(p,C_j) \;\; otherwise \end{cases}$

Notice that the second rule says that *p* can be assigned to a cluster even though not only the pattern itself, but also its generalized structure *p'* has never been encountered during the learning phase. Furthermore, since the same candidate *ct* can be extracted from different sentences and patterns p_i, the global confidence in a candidate is computed as:

$$weight(ct) = \frac{\max_{p_i \in C_{p_i}^*} (score(C_{p_i}^*) \times (1 - d'(p_i, C_{p_i}^*)) \times (1 + \ln(freq(ct))}{\max_{ct_n in \; EXP} weight(ct_n)}$$

The *max* function in the numerator selects the highest score obtained by any of the extracted patterns p_i that support *ct*, while the smoothing factor $(1 - d'(p, C_p^*))$ adjusts the weight of *ct* according to its membership in the selected cluster. Finally the factor $(1 + \ln(freq(ct)))$ increases the weight of *ct* according to the number of patterns that supported *ct*. The denominator is a normalizing factor over all the weights calculated for all the terms *t* in a given run of the algorithm. A threshold β is experimentally tuned such that a *ct* is returned only if $weight(ct) \geq \beta$.

Term matching is used during the *reinforcement* phase (step 3 of the algorithm), which is aimed at refining cluster weights, according to their precision and recall. During the cluster refinement phase, we take the set S_1 in *MC* and, separately for each element of a pair $(tt_i, nt_j) \in S_1$, we test the recall and precision of the patterns belonging to the various clusters, in order to adjust cluster weights. In fact certain patterns, e.g. "*or*", as in "*hypoglycemia or low blood sugar*" and "(", as in "*vomiting (emesis)*" are very frequent but have a low precision.

Given the terms in S_1 we test each pattern p_i in the following way: $n_{tp}(p_i)$ = number of true terms returned by p_i ; $n_{fp}(p_i)$ = number of false terms returned by p_i ; $n_{fn}(p_i)$ = number of true terms extracted by p_i but below the threshold β. We can then compute an additional weight for p_i that takes into account its performances: $weight_r(p_i) = (n_{tp}(p_i) + n_{fn}(p_i))$

and $weight^*(p_i) = weight(p_i) + weight_r(p_i)$

After this step, clusters weights are updated with the new pattern weights.

2.2 Evaluation

To test the algorithm we take S_2 and we perform term matching, using the adjusted clusters weights. We perform a six-fold cross evaluation, in which S_0, S_1 and S_2 are used interchangeably. Notice that in each run, the obtained clusters and weights can be different, since a different dataset is used to extract sentences from the Web. The global performances are averaged over all the runs.

For training, refining and testing purposes we use a set *MC* of 193 (tt, nt) pairs from Freebase.

To extract sentences we used the following web resources: Google snippets (up to the allowed query limits), Wikipedia, BMC BioMed Central Corpus[10], UKWaC British English web corpus[11].

During each run of a testing phase, we take a t_i from the dataset "playing the role" of S_2 and we try to extract from the previously listed web resources a set of correspondent partner terms, using the clusters and cluster weights learned in previous phases. We then compare them with the ground truth in S_2. Let *TT* the set of technical terms in the test set and $NT_i := \{nt_1^i, nt_2^i, nt_k^i\}$ the "true" set of naïve terms for each $tt_i \in TT$. To compute performances, we use standard measures such as *precision*, *recall* and *F-measure*, as well as the *mean reciprocal rank (MRR)*, a measure that prizes true positives if

[9] As for POS tagging, we use the Treetagger

[10] http://www.biomedcentral.com/about/datamining

[11] http://trac.sketchengine.co.uk/wiki/Corpora/UKWaC

they are top-ranked wrt the set of returned answers. *MRR* is defined as: $MRR = \frac{1}{|TT|} \sum_{nt^* \in NT_i \forall tt_i \in TT} \frac{1}{rank(nt^*)}$ where nt^* is a true positive for tt_i retrieved by the algorithm (e.g. $nt^* \in NT_i$), and $rank(nt^*)$ is the position of nt^* in the list returned by the algorithm. Since the test is repeated for any possible permutation of the three datasets S_0, S_1 and S_2, the performance is averaged over all the six experiments. The performance results are reported in Table 2 with $\alpha = 0.38$ and $(w^A, w^B, w^C) = (0.55, 0.30, 0.15)$. As expected, a higher threshold improves precision but reduces the recall. Furthermore, the high MRR shows that true positives are likely to receive a higher score wrt false positives, which is a desired property.

Since often for a technical term there might be many naïve terms, and Freebase is far from being complete, we asked two physicians (one is a co-author) to manually evaluate the extracted terms according to their expertise. In Table 3 the recall is computed considering the number of terms considered correct, both above and below the threshold. In the Table, *k-Fleiss* is the inter-annotator agreement[12]. The Table shows a higher precision, as expected, however there is quite a number of good terms below the threshold (recall is 0.49). In applications, the better strategy is to use no threshold and ask a physician to mark the correct terms. Given a disease under surveillance, this manual step is simple and requires few minutes, while there would be no easy way for a clinician to imagine, without the help of a text mining tool, the variety of expressions used by patients.

β	Precision	MMR	Recall
0	0.60	0.64	0.73
0.1	0.64	0.71	0.66
0.2	0.69	0.82	0.60

Table 2. Average system performance against golden-standard

β	Precision	Recall	F1	MMR	k-fleiss
0.2	0.76	0.49	0.59	0.74	0.53

Table 3. Manual Evaluation by domain experts

After the training phase, we selected the best performing clustering in the six experiments (namely, one with MRR=0,87) as the final model for extracting naïve medical language. We notice however that performances are not significantly variable and seem more related to the searched terms (i.e. whether they are more or less popular on the web) than to any of the clustering results.

3. Case Study On Five Syndromes

In this Section we apply the results of our algorithm to a case study of five common syndromes: influenza-like illness (with two sub-cases, ILI^{ECDC} and ILI^{FEVER}), common cold, allergic rhinitis, and gastroenteritis. Our clinical partnership used the results in (Rumoro et al., 2011) to create 5 queries, each testing for one of the following cases[13]: ILI^{ECDC}, ILI^{fever}, Gastroenteritis (GASTRO), allergic rhinitis (ALLERGY), common cold (COLD).

For example, the query for ILI^{ECDC} is:

((fever)OR(chills))OR(malaise)OR(headache)OR(myalgia))AND((cough)OR(pharyngitis)OR(dyspnea))

We used our algorithm to expand 17 symptom-related medical conditions (e.g. *rynhorrea, pharyngitis, myalgia, dyspnea, chills..*) mentioned in (Rumoro et al., 2011), and we retrieved an additional set of 62 naïve terms. Each symptom in a query was then expanded by adding its alternative retrieved terms. Using the available APIs[14], we collected a dataset of Twitter messages including at least one of the retrieved symptoms, from February 1st to May 6th 2013. To further extend the set of naïve terms, we used the patterns in Table 1 to extract additional candidates from our Twitter dataset. Overall, 29 additional terms are retrieved in this way.

Systematic keyword analysis has shown that being able to trace both technical and naïve terminology produces a much larger body of evidence. For example, as shown in Figure 1, on February 5th there have been 957 tweets including *watery eyes, bloodshot eyes*, etc, and 393 with *conjunctivitis* or *conjuntivitis*. Similarly, *pharyngitis or laryngitis* cumulated 47 tweets on the same day, while their correspondent set of naïve terms occurred 12,440 times.

[12] http://en.wikipedia.org/wiki/Fleiss'_kappa#Interpretation

[13] http://www.influenzanet.eu/en/results/?page=help

[14] https://dev.twitter.com/docs/streaming-apis

To evaluate the quality of retrieved tweets, for each of the five syndromes, we extract a set of 100 positive tweets (those matching the related query) and a random sample of 500 tweets not matching any query but including at least one symptom. Tweets are then examined by the physicians, to test whether they can be truly considered as reporting symptoms that match the considered case definition. Of course, it is impossible to verify if these users are truly affected by any of the 5 syndromes. The purpose is rather to assess the *confidence* we can have in our methodology as a mean to retrieve from Twitter messages that actually refer symptoms related to one of the analyzed syndromes. Examples of true positives, false positives and false negatives are:

***tp**: If this is the flu! I am going to be so pissed:/ **fever**, nausea, neck pain, **sore throat**, all this **coughing**..its back to bed!*

fp**: hate when people self diagnose no you haven't got 'depression' or '**tonsillitis**' you've had a bad day and a **sore throat

***fn**: #**puking** #**stomachache** #imsorry*

The results of the evaluation (reported in Table 4) show a remarkable precision, furthermore we found no false negatives in the random set of 500 tweets (the Recall estimate is then 1). We provide hereafter an analysis of error causes, including those that possibly could produce false negatives:

1. Tweets that report news or someone else's condition: most of these errors are eliminated by simply canceling re-tweets or tweets including an url, but some still survive, e.g. "*Symptoms of H1N1 are like regular flu symptoms and include fever, cough, sore throat, runny nose, body aches, headache, chills, and fatigue.*"
2. Negation: the presence of a negation in a tweet is not enough to determine if it is a negative case. For example: "***Not** bad. Throat infection, fever and flu all at once!*" is a true positive for ILI^{ECDC}, while: "***No** fever, diarrhea, abdominal pain. On Tamiflu now!*" is a false positive. More complex treatment of negation is needed to handle these cases, however they are a minority.
3. There are naïve expressions for a medical condition that were not extracted by our algorithm. These may cause both false positive and false negative. For example, looking at the data we found that *puking* is an additional synonym of *emesis (vomiting)*. The previously cited example of false negative is precisely due to this type of error, since one of the positive conditions for gastroenteritis is: *(emesis) AND (abdominal pain)* where *puking* is a naïve term for *emesis* and *stomachache* for *abdominal pain*.

	total tweets	**fp**	**Precision**
ILI^{ECDC}	270,503	3/100	0.97
ILI^{fever}	24,575	1/100	0.99
ALLERGY	42,062	0/100	1.00
COLD	145,657	1/100	0.99
GASTRO	102,980	15/100	0.85
Total	585,777	20/500	0.96

Table 4. Evaluation of the ILI-related case study

Figure 2 shows the trends of the analyzed syndromes. Note that, given the time span under analysis there is a high predominance of COLD and ILI^{ECDC}, while ALLERGY is growing since April, as expected.

Finally, we aim to correlate our data with those reported by the U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet), collected through the CDC Fluview website[15]. Figure 3 shows the time series for our Twitter messages, for Google Flu Trends, and the official ILINet data. All time series where smoothed by the *loss function* presented in (Cleveland and Devlin, 1988), to reduce the effect of daily fluctuations. The Pearson correlation Google/ILINet is 0.9927 and our geolocalized[16] time series ILI$^{ECDC-US}$ /ILINet is 0.9965.

4. Related Work

To the best of our knowledge (Elhadad and Sutaria, 2007) is the only paper in which the correspondence between technical and naïve terms is analyzed. The paper is however focused on pairing (tt_i, nt_j) terms when the set of technical and naïve terms is pre-determined, and defined in UMLS[17]. Another related area is synonym extraction, since naïve terms can be seen as synonyms or near synonyms of technical terms. In this area, most approaches are based on the so-called *distributional hypothesis*: words with similar contexts have a similar meaning. A very recent study on synonym extraction is described in (Henriksson et al., 2012), where random indexing and random permutation are applied to automatically extract variants of medical terms. We notice that performance is not

[15] http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

[16] http://www.jmir.org/2012/6/e156/

[17] *www.nlm.nih.gov/research/**umls**/*

very high: the best model for synonyms has a 0.42 recall while the precision is very low: 0.08 in the best experiment.

Semantic relation learning is also similar to our task at hand, since the objective is to identify sequences of words that imply a given relation between two terms, e.g. for causal relations: "*dengue fever is caused by which mosquito*". Patterns are either hand crafted, or they are automatically learned using some manually annotated set of sentences. Another difference among the various approaches is between fixed (or "hard") lexico-syntactic patterns, and generalized patterns, usually in the form of graphs. In her seminal work, Mart Hearst (1992) proposed a number of fixed lexical patterns to extract hypernyms from sentences, e.g. "X such as Y" . Snow et al (2004) first search sentences that contain two terms which are known to be in a taxonomic relation (term pairs are taken from WordNet), as we do for *tt-nt* pairs; then they parse the sentences, and automatically extract fixed patterns (features) from the parse trees. Finally, they train a hypernym classifer based on these features. The approach requires the annotation of a possibly very large set of sentence fragments to train the classifier, and final performance is not so high. Cui et al. (2007) propose the use of probabilistic lexico-semantic patterns, called soft patterns, to identify definitional sentences. Finally, Navigli and Velardi (2010) use word-class lattices (WCL) to identify definitional sentences, starting from a large dataset of annotated definitions, where the *definiendum* and *definiens* tems have been manually annotated. Like for soft matching, WCL provide a generalization of patterns, where nodes of a lattice are either words or part of speech tags. Our work builds on WCL's idea of replacing words in a sentence fragment with POS, while keeping nouns and functional words. The subsequent generalization steps are different, since we use semantic categories and pattern clustering rather than lattices, and furthermore, no manual annotation is needed.

Considering the literature on the use of web data for disease prediction, the most relevant work related to our study is reported in (Ginsberg et al., 2009). In this work the authors fit a linear model for predicting ILI epidemics using query volumes data and historical data from the CDC's US Influenza Sentinel Provider Surveillance Network. To automatically obtain relevant keywords they use a set of 5 years, 50 millions Google web search queries. To select the appropriate keywords from these queries, they perform a correlation study for each query, to test if it models accurately the CDC ILI data in nine regions. This study is certainly more accurate wrt previous similar works that use few manually defined keywords (Althouse et al., 2011), such as *flu* and *influenza*. However, first, the algorithm depends on the availability of critical resources: web query logs are a kind of data which is *not freely available*. Our algorithm instead, once the model is learned, allows it to extract the relevant keywords automatically (possibly with a quick manual post-editing), for any disease or symptom. Second, given the large amount of initial queries (50 millions), keyword selection and correlation estimation for each possible keyword becomes a very demanding task, and in principle, it should be repeated for any disease under surveillance, on continuously updated query log data, since new keywords may appear (e.g. this year the predominant flu strain is H3N2 and still lacks a nickname, previous names have been *swine flu, bird flu*, etc.). Third, measuring query search volumes has the problems that we outlined in the introduction (ambiguity, sensitivity to external events): blogs and forums provide keywords in contexts, fostering more interesting types of analyses, as shown in our ILI case study. Another recent work (Lamb et al. 2013) separates tweets reporting infection (*flu*) from those expressing concerns and fear ("*a little worried about flu epidemic*!"). To automatically separate these tweets, the authors use a log-linear model and a set of fine-grained manually identified features (e.g. expressions of concern, such as *afraid, worried, scared*). This method, which is complementary to our symptom-driven technique, is reported to obtain 0.9897 Pearson correlation with ILINET on a 2009 sample, but only 0.7897 in a 2011 sample (when also Google Flu obtained 0.8829).

5. Conclusions

Overall, the results of this study show that knowledge of patient's language fosters the exploitation of social media not only to predict disease outbreaks, but also to classify patient symptoms in more fine-grained cases. Our methodology is more powerful vrs. e.g. Google Flu Trends, since it may help estimating the seriousness of any disease outbreak, the incidence of individual symptoms (e.g. *cephalgia* was a predominant flu symptom this year), to

classify an illness in sub-cases (ILI vrs common cold), to detect frequently – and possibly unexpected- co-occurring symptoms, etc. For the sake of space, we reported here only a fragment of our findings.

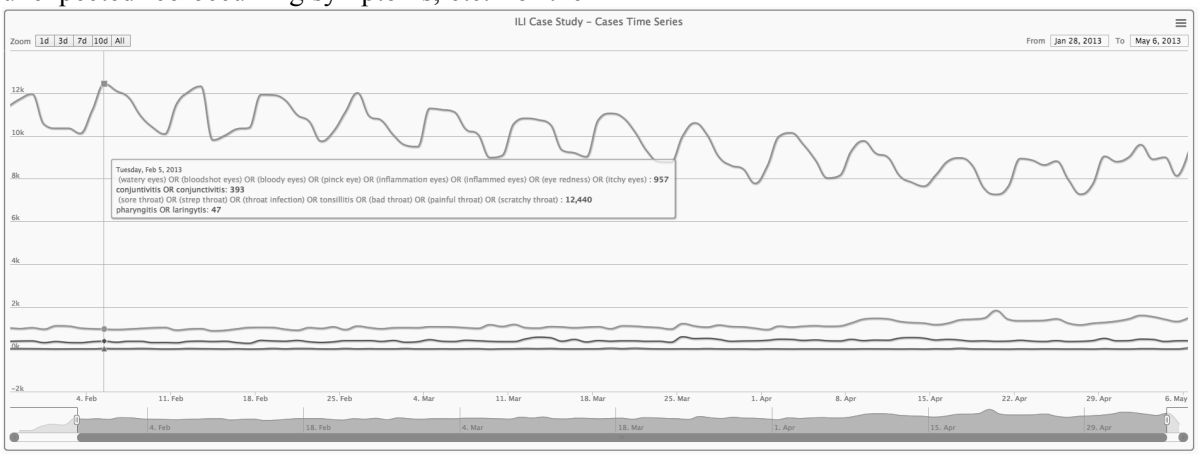

Figure 1. Total traffic for *laryngitis,pharyngitis* and correspondent naive terms, and for *conjuntivitis* and correspondent naive terms.

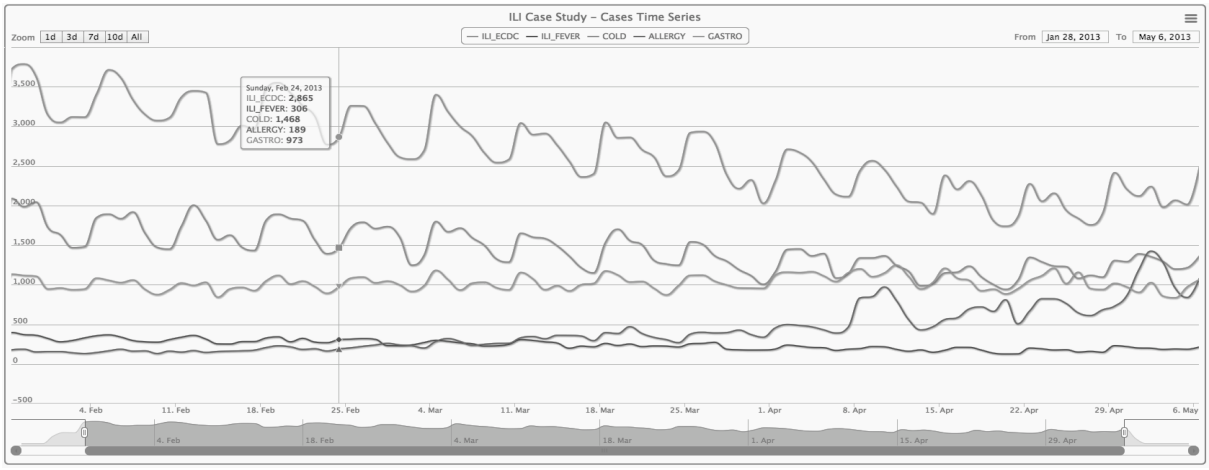

Figure 2. Total traffic for the five analyzed syndromes

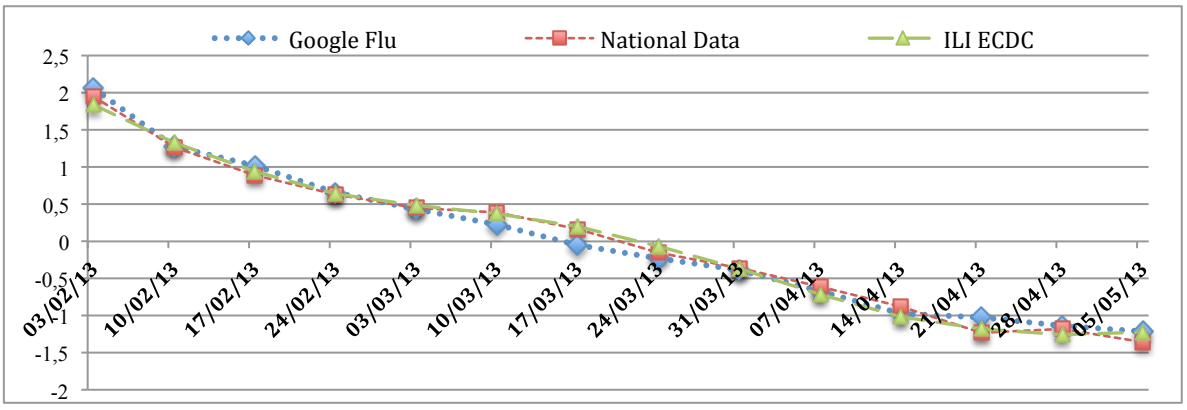

Figure 3. Correlation among Google Flu Trends, ILINet official data, and ILI^{ECDC} (US data)

References

B.M. Althouse, Y.Y. Ng, D.A.T. Cummings, (2011) *Prediction of Dengue Incidence Using Search Query Surveillance.* PLoS Negl Trop Dis 5(8)

B. Berendt (2011) *Text Mining for News and Blogs Analysis,* Encyclopedia of Machine Learning, Springer Science+ Business Media, LLC, 10.1007/978-0-387-30164-8_827

W. S. Cleveland and S. J. Devlin (1988), *Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting*, Journal of the American Statistical Association, Vol. 83, No. 403 (Sep., 1988), pp. 596-610, American Statistical Association.

A. M. Cohen and W. R. Hersh (2005) *A Survey of current work in biomedical text mining*, Henry Stuart Publications *1467-5463, Briefings in Informatics*, Vol 6. NO 1. 57–71. March 2005

C. D. Corley (2009) *Social Network Simulation and mining social media to advance epidemiology*, PhD dissertation, University of North Texas

H. Cui, M. Kan, and T.Chua. (2007) *Soft pattern matching models for definitional question answering*, ACM Transactions on Information Systems (TOIS), vol. 25 n. 8

M. R. Dahm, (2011) *Exploring Perception and Use of Everyday Language and Medical Terminology* among International Medical Graduates in a Medical ESP Course in Australia, in: English for Specific Purposes, Elsevier v. 30 n. 3 pp186-197, Jul 2011

N. Elhadad and K. Sutraria (2007) *Mining a Lexicon of Technical Terms and Lay Equivalents.* Proceedings of the Workshop on BioNLP 2007

P. von Etter, S. Huttunen, A. Vihavainen, M. Vuorinen and R. Yangarber (2010) *Assessment of Utility in Web Mining for the Domain of Public Health*, Proc. of NAACL HLT 2010, pp 29-37

G. Eysenbach (2006) *Infodemiology: tracking flu-related searches on the web for syndromic surveillance.* AMIA Annual Symp Proc. pp 244–8.

J. Ginsberg, Mohebbi MH, Patel RS, Brammer L, Smolinski MS, et al. (2009) *Detecting influenza epidemics using search engine query data.* Nature 457:1012–4.

A. Henriksson, H. Moen, M. Skeppstedt,AM Eklund, V. Daudarvicius and M. Hassel (2012) *Synonym Extraction od Medical Terms from Clinical Text Using Combinations of Word Space Models*, in Proc. of the 5th International Symposium on Semantic Mining in Biomedicine (SMBM 2012)

M. Hearst (1992) *Automatic acquisition of hyponyms from large text corpora*, Proc. of the 14th International Conference on Computational Linguistics, Nantes , France.

A. K. Jain, (2010) *Data clustering: 50 years beyond K -means.* Pattern Recognition Letters , 31: 651–666, 2010

Alex Lamb, Michael J. Paul, Mark Dredze (2013). Separating Fact from Fear: Tracking Flu Infections on Twitter. North American Chapter of the Association for Computational Linguistics (NAACL), 2013.

Molina HealthCare and California Academy of Family Physicians, (2004) *Medical Jargon and Clear Communication*, CAFP's California Bureau of Registered Nursing Provider #1809

Roberto Navigli and Paola Velardi (2010*) LearningWord-Class Lattices for Definition and Hypernym Extraction*, Proc. of ACL 2010

Michael J. Paul and Mark Dredze (2011) *You Are What You Tweet: Analyzing Twitter for Public Health*, In the proceedings of the *5th International AAAI Conference on Weblogs and Social Media (ICWSM 2011)*, Barcelona, Spain. July 2011.

Dino Rumoro , Shital Shah , Julio Silva , Marilyn Hallock , Gillian Gibbs and Michael Waddell (2011) *Case definition for real-time surveillance of influenza-like illness*, Emerging Health Threats Journal 2011, **4**: 11123

R. Snow, D. Jurafsky, and A. Y. Ng. (2004) *Learning syntactic patterns for automatic hypernym discovery*. In Proceedings of Advances in Neural Information Processing Systems, pages 1297–1304

N. Zhong , Y. Li and S. Wu (2012) *Effective Pattern Discovery for Text Mining*, IEEE Transaction on Knowledge and Data Engineering, vol 24, n. 1 , January 2012

D. Xu, Y. Liu, M. Zhang, S. Ma, A. Ciu and L. Ru (2011*) Predicting Epidemic Tendency through Search Behaviour Analysis*, Proc. of 22nd IJCAI