Tracking the Diffusion of Named Entities

TBD

Abstract

To do

Index Terms

To do

I. Introduction

The aim of this paper is to understand how named entities *emerge* and *spread* through social media based discourse. We are interested in exploring the following research questions:

1) **RQ1:** How can we accurately detect named entities in social media based discourse, given its myriad formats, often informal vernacular, and inherent noise (e.g. misspellings, abbreviations, etc.)?
2) **RQ2:** Under what conditions do entity mentions diffuse through discourse? And when are people *most likely* to be influenced into then discussing entities?
3) **RQ3:** How can we predict the discussion of certain named entities and who will begin talking about them?

II. Datasets

For this research we will use the following datasets:

1) Reddit data – download and access all of the data from the full dump.[1]
2) CoNLL 2003 data – a corpus of newswire texts, annotated for named entity chunks and types. This describes where entity mentions are in the text, including locations, organisations, and person mentions.
3) Twitter data; unannotated – we have a large corpus of English tweets that we can use here.
4) Twitter data; annotated – there are two datasets annotated with named entities. These are from Ritter's 2011 EMNLP paper, and the W-NUT 2015 shared task.

III. Research Stages

A. Stage 0: Data Preparation and NER

-To do:

–Annotate corpora with detected entities using basic typing of: person, location, organisation

–Run NER software over dataset and validate accuracy of this (using basic measures)

–Run NER over entire dataset to extract entities

[1] https://archive.org/details/2015_reddit_comments_corpus

B. Stage 1: Exploratory Analysis

-To do:

–Plot relative frequency distribution as a function of time for named entities, and characterise the *shape* of the entities

–Apply lifecycle model to profile users' NER citations over time and investigate how users' profiles are influenced by global, community, and prior behaviour dynamics

C. Stage 2: Diffusion Analysis

-To do:

–Model the spread of named entities through user profiles (could use multivariate diffusion models here)

D. Stage 3: Forecasting

-To do:

–Implement models to forecast if a user will mention an entity and who that will be (hard!)

IV. Related work

twitter info propagation

reddit compared to other OSNs: "Lifespan and propagation of information in On-line Social Networks: A case study based on Reddit" http://www.sciencedirect.com/science/article/pii/S1084804515001307

network structure of reddit: "Navigating the massive world of reddit: using backbone networks to map user interests in social media" https://peerj.com/articles/cs-4/

twitter nlp

V. Data preparation and NER

To conduct our study, we need to convert 140GB of compressed Reddit posts into a set of interlinked and time-ordered conversations and the entities mentioned in each of them. This provides a number of sub-challenges: sampling of the Reddit data, creating a linked series of conversations, and picking out entity mentions in this text type. Reddit data is largely unexplored in the NLP community, despite the large volume of it and the especially rich metadata. This poses additional challenges: certainly, given the lack of work on Reddit text, there are no annotated datasets available yet, so supervised in-domain work is not directly possible yet. Additionally, the datasets are large, which makes it important to choose a good subset of data on which to do prototyping and development, in order to keep research cycles short. The result that we come to at the end of this stage is a rich dataset for tracking entity and concept diffusion within and across communities.

The Reddit dataset [1] is comprised of a sequence of comments, with one JSON record for each one. These are ordered temporally. Reddit itself is roughly similar to a forum, where top-level divisions are made by topic. Within each topic, or *subreddit*, there are posts, which begin with either a short piece of text or a link to an external resource – typically an image, video, or interesting article. Users then may publish comments for each post, and

Fig. 1. Example discussion around a Reddit post.

reply to each others' comments. This leads to a threaded discussion, centred on a particular topic, with a hierarchical comment structure (see Figure V).

A. *Subreddit extraction*

- subreddit extraction

top lists

describe top lists site

pick 100

extract these over whole sample

figures for raw dataset (# per year; total volume over time; volume-rank graph of subreddits; possible ref to appx)

B. *NER for Reddit*

– where is NER first mentioned and defined?

We model micro-topics in conversation as entity mentions. This allows tracking of topics at maximally fine granularity, looking at each user's interests at a low level, as opposed to monitoring broader topics such as "consumer electronics", "politics" and so on. In fact, these broader topics are already explicitly annotated by means of the subreddit topics.

Entity mentions are extracted through named entity recognition. Generally, this task aims to detect the boundaries of certain kinds of entities within a certain piece of text. In this instance, we tokenise text, splitting it into

sentences using the Punkt tokeniser [2], and subsequently word-sized chunks, using the `twokenize` tool with some adaptations [3]. This tool performs Penn Treebank-style tokenisation, a common standard, with some specific adaptations to enable it to handle the noise present in user-generated text. After this, we take a structured prediction approach to deciding which tokens in each sentence are part of an entity, and possibly the type of the entity. Finally, we concatenate entity tokens, and use these to build a list of entity mentions in any given input text. For example, given the input comment from the source JSON:

"body": "There are still some really good fighters on this card. Conor McGregor is on the card and so is Gunnar Nelson."

The following output entities should be collected:

"entity_texts": ["Conor McGregor", "Gunnar Nelson"]

Typically, many NER systems take a supervised approach; that is, they use data labelled by humans as training data, from which features are extracted to form training instances for a machine learning algorithm. However, NLP systems can be hard to transfer between text types; for example, NER systems for newswire might reach about 89% F1 on news articles, but only around 40% on tweets (a form of user-generated content), as found in [4]. One approach to overcoming this performance drop when changing text type is to train over a blend of text types. For example, Ritter [5] used both IRC[2] and newswire data when developing a part-of-speech tagger for tweets, as well as an unsupervised language model from the target text type. This led to strong performance improvements. We follow a similar approach, using a blend of NE-annotated corpora from both newswire and tweets. The newswire data is drawn from the CoNLL-2003 evaluation task set [6]; the twitter data is from Ritter's early experiments and also the W-NUT 2015 shared task [5], [7].

We start using structured predicting in the form of a CRF to label whole sentences at a time. For features, we use a fairly classical set, and add some unsupervised word representations to this. Our base features are:

•

In addition, we induce Brown clusters [8] and use these as word representations [9]. Brown clustering is a form of hierarchical agglomerative hard clustering, using average mutual information as its objective function. It takes as input a corpus, in the form of a sequence of words, and in its generalised form [10], a single hyperparameter: the size of its active set a. The result is a sequence of binary merges, describing the set membership of each word type in the corpus as the merges progress. For each single word type, therefore, the path to a destination cluster can be described as a bitstring, which details the sequence of binary merges taken. The zero-length bitstring describes the situation at the top of the hierarchy, where there is one class.

These bitstrings are typically converted to features by shearing. This involves only examining the first n bits of a bitstring. However, shearing does not maximise the information preserved in the representation – sub-clusterings at many levels are lost. We therefore experiment with a new method of feature extraction form Brown clusters, which we call *provenance-based* feature extraction. We take the cluster identifier at every level, tracing the provenance of a terminal word cluster all the way to the root cluster (which contains all word types). This preserves the entire set

[2]Internet Relay Chat – informal internet conversation text

membership of any given term, throughout the induced hierarchical clustering. As an example, if the term *fishing* has bitstring `1100101`, the following text features are generated: `1`, `11`, `110`, `1100`, `11001`, `110010`, `1100101`. For contrast, if the typical bit depths of 4, 6, 10 and 20 [11] were chosen, only the following features would be generated: `1100`, `110010`, `1100101`. As a result of taking all directly-relevant features in the merge list, the lossy nature of shearing-based feature extraction from Brown clusters is avoided.

Feature extraction, training, classification and JSON annotation are all performed using an entity recognition toolkit (https://github.com/leondz/entity_recognition [12]), with custom extractors.

C. Tuning entity recognition

Entity recognition needs to be tuned to fit Reddit data well. There are a number of parameters in our training data balance, feature extraction, and objective function that all reflect the nature of the data and the task at hand. We present our method for estimating of these factors, and intrinsic NER evaluation.

In terms of evaluation, we prefer recall over precision. Over the large dataset, spurious entities are likely to be those that are not seen very often or have an unusual pattern. This suggests that there will be great variation in their surface forms, leaving them in the long tail of discovered entities. As the majority of our work looks at the more frequent patterns, these are less likely to have an impact. Conversely, recall reveals how well the extraction is performing, and it important to track a range of entities. In addition, recall has always been more challenging to achieve in social media texts than recall [5], [4].

To capture this balance, we take F-scores with $\beta = 2$. Given precision P and recall R, typically an F-score is drawn from F_β with $\beta = 1$.

$$F_\beta = (1 + \beta^2)\frac{PR}{(\beta^2 P) + R} \tag{1}$$

When $\beta = 1$, precision and recall are balanced in a harmonic mean, e.g. F1-score. That is, false positives and false negatives impact results equally. To score away from false positives, i.e. wrong cues, we set $\beta = 2$.

Our approach here is to tune an entity recogniser with reference to a dataset that matches the target text type. We draw this development set from Reddit posts, using comments encountered during our work that appear to have missing or spurious annotations. These are then isolated, tokenised, and manually annotated. Our annotation format follows [5] in using the Freebase top-level entity type inventory, but only uses the chunking information, as nothing further than this is needed: only the surface forms of entities. In total, we identified and annotated 3 708 tokens of Reddit data, including 149 entity chunks. This comprised our development set, which was used to tune a variety of parameters in our approach.

Firstly, we tuned our word representations. Specifically, we needed to estimate the number of Brown clusters C to use in feature extraction. We also then used this to determine what blend of social, Reddit or news data gave best results in unsupervised feature generation and extraction. To tune the value for C, we examined a similar scenario with similar dataset sizes, and estimated an optimum. We noted that in prior work [13], entity recognition performance with decomposed class prefixes – similar to the provenance-based features we propose here – peak at

around $C = 2500$ for corpora of 16k tokens, $C = 5000$ for corpora of 32k tokens, and at higher values for larger datasets. As General Brown clustering is dependent on the number of types and the size of the active set a, and results are unreliable with $a > C$, we set $C = a = 4000$. We then experiment with combinations of newswire, Twitter and Reddit data. Brown clusters are extracted using the generalised-brown package [14]. Results are given in Table **??**.

– results of brown cluster tuning

In addition, we draw supervised data for multiple datasets in order to approximate the Reddit text type. We take data from Reddit, taking the union of corpora used in previous work that follow the Freebase ten-class entity scheme [5], [7]. This scheme gives broader coverage than e.g. the three-class ACE named entity scheme. For newswire, we use the Reuters RCV1 corpus annotations that were part of the CoNLL-2003 shared task [6]. Classes are removed before training, making this just a chunking task. We compare against Stanford NER [15] as a baseline. Results for different balances of training data are given in Table **??**

– analysis of cross-genre ne chunking performance

Note that while some figures seem low when compared to typical newswire level performance, the toolkit used is high-ranking, state-of-the-art research software, coming third in the 2015 W-NUT challenge for entity chunking over tweets. The task is simply difficult; Twitter NER recall has always been low. In addition, it is a generally consistent finding that generalising NER systems beyond newswire is not yet well understood; systems that perform very well on this text type (e.g. F1 of 0.89 from Stanford NER [4]) can often score very poorly on social media content (F1 of 0.41). This may be due to overfitting of tools to newswire over time, due to community challenges, dataset in just one type, extra custom rules adapting to formal news text, or other things – but this is beyond the scope of this paper. We do note that our approach uses largely unsupervised feature extraction and performs better than on other social media corpora, also beating the Stanford NER system, in this first attempt at named entity chunking for Reddit.

VI. Entity Diffusion

In this section we now move on to examining how the recognised entities emerge and *diffuse* through the analysed subreddits. As per prior work, one of the first things that we can inspect is the *shape* of entity mention cascades: that is, the patterns of diffusion that such entities exhibit when cited in conversation chains. We begin by explaining how such patterns are derived, before then moving on to showing that patterns emerge.

A. Entity Mention Cascades

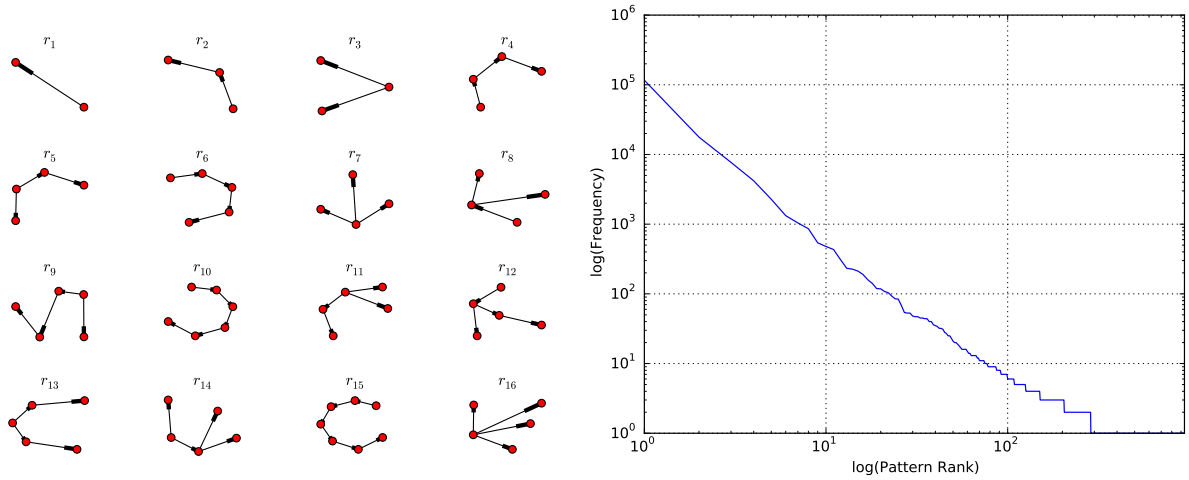
Prior work by Leskovec et al. [16] examined the shapes of hyperlink cascades through the blogosphere; in doing so, investigating that patterns appear in terms of the diffusion of links. We follow a similar process here, however we instead inspect the emergence of entities in conversation chains in reddit. We first make the following explicit.

Definition 1. *(Entity Cascade) A cascade of* $< p_i, pj > \in C_e$ *of an entity* $e \in E$ *occurs when two or more posts citing the entity are chained together in a reply graph. Hence:* $C_e = \{< p_i, p_j >: p_i \rightarrow p_j \in R, cites(p_1) = cites(p_j) = e\}$.

Our goal is to derive all entity cascades for each entity in our analysis, and then examine how the shapes and sizes of these cascades differ. To gather each entity's cascades, we retrieved all subreddit posts that contained a given entity. For each post ($p \in P_e$), we then recovered the reply-chain that that entity appeared within - this was performed by going *up* the reply chain from p to its parent post (i.e. the post that p was replying to) and *down* the reply chain by getting the posts that replied to p. When iterating through the posts, if we came across a post that replied to another post in an existing chain then that post was added to the chain. We only maintained posts within the chain that cited the entity in question: this produced entity cascades where each consecutive post in the chain mentions the entity - we refer to this as *strict cascade derivation*, as we do not consider posts higher-up or lower-down the reply chain that cite the entity yet are connected by a non-entity citing post. [3]

This process produces, in essence, a collection of cascade graphs for each entity, each of which may have isomorphic shapes yet contain different node labels (i.e. different post ids). We reduced each entity's cascade graph collection down to a frequency distribution of the *canonical form* of each graph using Cordella at al.'s [17] graph isomorphism approach. A further reduction was run to compile a frequency distribution of the cascade shapes across all entities. Fig. 2 show both the top-20 entity cascade shapes on the left (Fig. 2(a)) and the ranking of the patterns' frequencies on a log-log scale (Fig. 2(b)). Upon inspection, one thing becomes immediately apparent: entity cascades are shallow and short at the top-3 ranks, however after this position we start to see chains of discussions as being popular which are deeper and narrow. This result contrasts somewhat to prior work [16] where cascades of hyperlinks between blogs were shallower in depth yet wider - in terms of the breadth of diffusion at the first level from the seed. The ranking of the patterns follows a general power-law distribution where a small section of patterns (i.e. the top-20) are seen most often - this is somewhat expected as it would be very rare for an entity to be cited in a long thread with many branching reply-chains.

[3]The Python code for deriving these chains is available at the following URL: https://github.com/mattroweshow/NER-Diff-Paper



(a) Top-20 Cascade Shapes

(b) Cascade Shape Rank Distribution

Fig. 2. The top-20 cascade shapes are generally deep and narrow with little branching (Fig. 2(a), while the cascade shape rank follows a power-law distribution (Fig. 2(b).

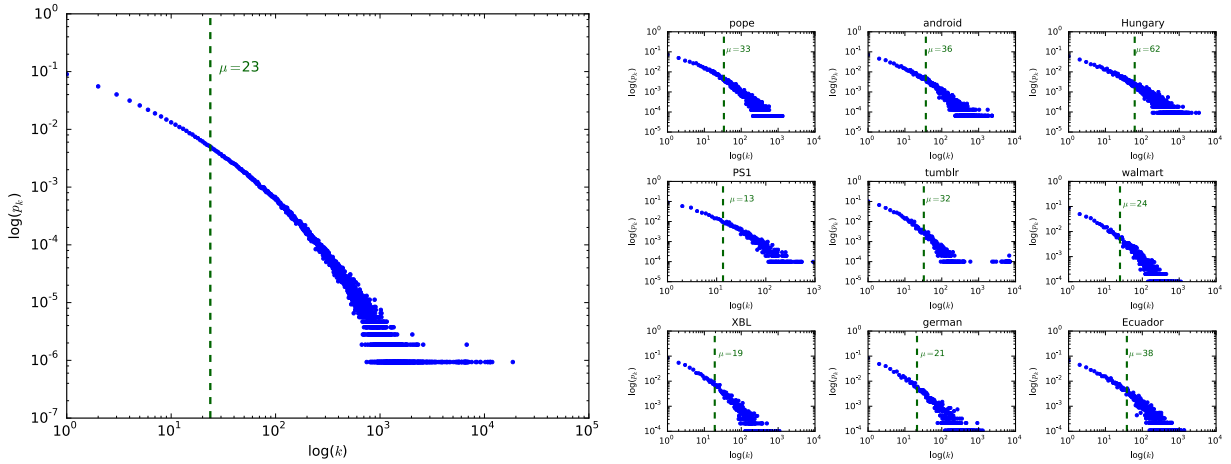
B. Entity Adoption Post-$(k-1)^{th}$ Exposure

Inspection of the shape of entity cascades through Reddit discussion threads reveals some interesting traits, suggesting that an entity is likely to be a adopted if it has been discussed *a priori* within the same discussion thread. One natural question that emerges from this is the extent to which exposures to entities play a role in actually adopting (i.e. citing) the entity in question. To investigate the relationship between exposures and adoptions, we sampled the top-500 entities from our whole annotated dataset and calculated the probability of a user adopting an entity after being *exposed* to the entity k times.

Definition 2. *(Exposure) A user u is exposed to an entity e at time t if a given post $p \in P^{\Gamma(u,t}$ authored by a neighbour of $v \in \Gamma(u)$ contains the entity e, where neighbours of u are those users that he interacted within prior to t.*

Based on this definition we iterated through all posts chronologically that cited a given entity, and if the post was the first time that a user cited the entity (i.e. he/she was not *activated*) then we counted how many *exposures* the user had received prior to the time of the post - logging this as k. Fig. 3(a) present the *overall* plot of the probability (i.e. relative frequency) of users adopting an entity after k exposures to the entity. Immediately, one can note that the mode of this distribution is at 0 and that the mean if $k = 23$: this implies that users are most likely to actually cite an entity without having been exposed to it, in fact $P(adoption) \rightarrow 0, k \rightarrow \infty$. We are somewhat guarded in *generalising* from this result, as our experimental setup here - given the scale of the data we are playing with and the tractability of annotating the entirety of Reddit - does result in only a fraction of Reddit being annotated with entities. Hence, it is possible that entities emerge from other subreddits, yet we are unable to capture this at present - our future work makes suggestions as to how this effect can be validated.

The second plot below (Fig. 3(b)) shows a sample of 9 entities' adoption-exposure distributions, all of which have similar shapes (with a mode at 0) and a heavy-tail. Variance exists, however, in the means that these distributions have, for instance the entity *PS1*[4] has a much lower mean than the entity *Hungary* suggesting that users require less stimulation to discuss the former than the latter. The nature of how and why the distributions differ is something that requires further investigation.



(a) Exposure-Adoption Global Distribution

(b) Sampled Entity Exposure-Adoption Distributions

Fig. 3. The probability of a user adopting an entity as a function of k prior exposures to the entity has a heavy-tailed distribution (Fig. 3(a)) that is consistent across all entities, including a sample of 9 random entities (Fig. 3(b)).

C. Global Threshold Diffusion Model

We now move on to forecasting the diffusion of entities across Reddit. For this, we use a modified implementation of Goyal et al.'s general threshold model [18] to parallelise the computation of the model. The core principle of the model is that one can calculate the probability of a user (u) adopting an entity (e) based on how his neighbours ($v \in \Gamma(u)$) have influenced him previously. Hence, the probability of u adopting an entity is calculated as follows:

$$p_u\big(\Gamma(u)\big) = 1 - \prod_{v \in \Gamma(u)} \big(1 - p_{v,u}\big) \tag{2}$$

In Goyal et al.'s prior framework, the probability of influence ($p_{v,u}$) of v on u is based upon the maximum likelihood estimate of a single Bernoulli trial. An entity propagation occurs between v to u when when the latter cites e after being exposed to it by the former (as per Definition 2), hence a count of how many entities propagate between v and u can be recorded in E_{v2u}. From this, the influence probability between v and u based on such *propagation* can be calculated as follows, where E_v is how many times v has cited an entity:

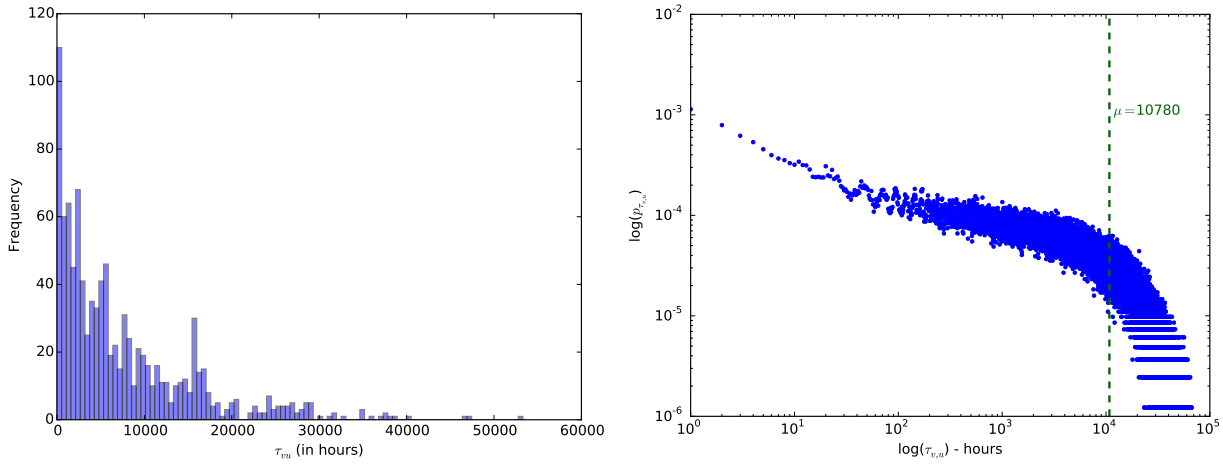
$$p^E_{v,u} = \frac{E_{v2u}}{E_v} \tag{3}$$

[4]Denoting the original Playstation video-games console.

The authors explain how there are two variants of this calculation: (i) a static Bernoulli random trial where Equation 3 is calculated from the training set (ignoring time), and: (ii) a discrete time Bernoulli random trial where counts are only placed within E_{v2u} and E_v if they the citation of an entity is within a discrete time interval, that is: if the time that u adopts an entity e is given by time t_u then E_{v2u} and E_v are composed from the entity posts of v which each have time $t_v \in [t_u - \tau_{v,u}, t_u)$, where $\tau_{v,u}$ is derived as follows (only considering $v, u \in U$ (set of all users) if u has contacted v prior to t_u:

$$\tau_{v,u} = \frac{\sum_{e \in E} (t_u(e) - t_v(e)}{E_{v2u}} \tag{4}$$

Fig. 4(a) shows the binned distribution of the $\tau_{v,u}$ values (reduced from milliseconds to hours to ease legibility): one can note how the distribution has a right skew with the mode of the distribution (roughly) being at one hour - that is, the majority of *influence* from one person to another occurs within one hour, this then gradually curtails off with fewer people having larger *influence windows*. The log-log plot of the relative frequency distribution (Fig. 4(b)) shows the *heavy-tail* property of the distribution, and that the mean window width is $10,780$ hours (≈ 449 days ≈ 1.2 years), thus suggesting a degree of *stickiness* in the Reddit communities where people remain for long periods.



(a) Binned Frequency Distribution of $\tau_{v,u}$ values

(b) Log-log Plot of the Binned $\tau_{v,u}$ distribution

Fig. 4. The influence window ($\tau_{v,u}$) between two arbitrary users characterises the average time for an entity to propagate from v to u. In hours, this value has a *right-skew* (Fig. 4(a)), while the log-log plot (Fig. 4(b)) of the relative frequency distribution demonstrates the heavy-tail nature of the distribution with a mean of $10,780$ hours (≈ 449 days ≈ 1.2 years).

1) Additional Influence Dynamics: The neat formulation of the general threshold model, and the monotonic-submodular nature of the probability of adoption function ($p_u(\Gamma(u))$, means that we can vary the mechanism by which we derive the *influence probability* ($p_{v,u}$) between two user v and u to test for different influence effects. Our contribution here is to test for the influence of *prior interactions* and *community homophily* using the general threshold model. To compute the influence probability based on interactions, we derive $p^I_{v,u}$ as follows:

$$p^I_{v,u} = \frac{|\{p_u : p_v \in P_v, p_u \in P_u, p_u \rightarrow p_v\}|}{|\{p_u : p_u \in P_u, p_u \rightarrow .\}|} \tag{5}$$

While the influence probability based upon community homophily ($p^C_{v,u}$ is derived as follows:

$$p^C_{v,u} = \frac{|C_u \cap C_v|}{|C_u \cup C_v|} \tag{6}$$

We calculate the same two variants as the entity-propagation influence probability as above (static Bernoulli, and discrete-time Bernoulli) for these influence probabilities by only considering interactions between u and v that fall within $[t_u - \tau_{v,u}, t_u)$ (for interactions-based) and posts within subreddits by u and v that were made within $[t_u - \tau_{v,u}, t_u)$ (for community-homophily).

In order to eliminate bias in our below experiments - when we attempt to forecast entity adoption for users - we divided the top-500 entities into an 80%:20% split for training and testing respectively. Then, for the above influence probabilities (entity-propagations, interactions-based, community-homophily) we used different strategies for their calculation. For the entity-propagation influence probability ($p^E_u(S)$) we used the training segment to calculate the values of E_{v2u} and E_v, and also $\tau_{v,u}$ - for all pairs of interacting users within the training segment - this follows the experimental setup of [18]. One thing that is somewhat limited about this approach, is that we are observing future effects when calculating E_{v2u} and E_v that we take forward into our experiments, as we observe how influence has occurred between users prior to an adoption happening. To some extent, this is somewhat unavoidable in the context of the dataset splitting as $\tau_{v,u}$ must be calculated somehow - an alternative for future work here is to use a fixed time-split and use the first 80% of entity-posts for training and the rest for testing - this does somewhat limited the generalisation capability however as new entities that appear in the latter portion would have limited information to learn their patterns of diffusion from.

D. Experiments

As mentioned above, one of our goals here is to *forecast* the adoption of entities by users as their spread through Reddit. To this end, we used an experimental setup that induces the joint probability function (Eq. 2) for the users citing entities within the test set by tracking their probability of adopting a given entity as a function of their neighbours influence. Our goal therefore is to examine which of the above influence probability functions are best suited to predicting adoptions. We now explain our experimental setup before moving on to present the results of such models.

1) Experimental Setup: Using the 100 randomly sampled subreddits and running the above Named Entity Recogniser over these subreddits' posts, resulted in a total of XXX posts in our dataset (using only those from the 100 subreddits) written by 4,139,814 users. Of those users we had 2,745,311 user-to-user edges derived from XXX interactions.

a) Deriving Adoption Probabilities: In order to test which model was best (from above) we took the entities within the test set, and ran the following process: we chronologically ordered each entities' posts and then iterated through the posts set one-by-one. For each post's author (v) we then checked if he had been *activated* before - i.e.

had he cited the entity? - if not, then this would be first time he had cited e. If this was the case then we retrieved the prior interactions that the user had had and calculated (for each prior neighbour - $u \in \Gamma(v, t_u)$) the probability of influence between v and u using the above influence probability variants (e.g. interactions-based with static Bernoulli setting). We then updated the probability of adoption of u. By iterating through the set of time-ordered posts we maintained adoption-outcome tuples of the form $< u, p_u, r_u >$ where $r_u \in \{0, 1\}$ denoting whether the user ultimately adopted the entity e or not. Our evaluation of the models used these tuples to calculate the area under the Receiver Operator Characteristic curve, aiming to achieve a value of 1 (for perfect prediction - with 0.5 being the random model baseline.

b) Parallelising Processing: As we are working with *big data* here, we made two efforts to parallelise the processes above - induction of adoption probability over the test set entities. First, all of the data used for the experiments (timestamped interactions between users, entity posts, post details, E_{v2u} values, τ_{v2u} values) was uploaded into HBase[5] tables ensuring that we could *quickly* access the data using time-specific queries (e.g. *return all interactions where u is the source before 12th January 2010*). Second, we used Apache Spark[6] to parallelise the per-entity diffusion processes. This was performed by loading the names of the test entities into HDFS and then forcing Spark to partition the entity list into at least 30 partitions. Each partition was then iterated over and the the above test process run: (i) retrieving time-ordered entity posts from HBase, (ii) iterating over the post set, (iii) retrieving per-user interactions prior to the time of a given post, (iv) calculating the pairwise influence probabilities. The final calculated probability of adoption for each user (u) together with the label of whether they adopted the entity or not were recorded in a separate HBase table of results.

Unfortunately, due to the use of sample of 100 of the top-500 entities in our experiments, iteration over the the time-ordered post set requires an expensive sequential scan - which cannot be avoided. That said, we were able to add a second level of parallelism however, given the sub modular and monotonic nature of the joint probability as follows. Calculation of the probability of e being adopted by u is derived from Eq. 2, and is calculated from the prior neighbours of u before adoption. Now, as this function is sub modular and monotonic, can *update* the probability of adoption given a new neighbour's (v) influence probability having being calculated as:

$$p_s(\Gamma(u) \cup p_{v,u}) = p_s(\Gamma(u)) + (1 - p_s(\Gamma(u))) * p_{v,u} \tag{7}$$

Now, our additional layer of parallelism occurs by *multithreading* the calculation of the influence probabilities between v and each of his neighbours $u \in \Gamma(v)$, thus we calculate these pairwise influence probabilities in parallel and then update $p_u(\Gamma(u)) : \forall u \in \Gamma(v)$.[7] In sum: our first layer of parallelism occurs by splitting the processing up per entity (using Apache Spark to process a partition of entities in parallel); while our second layer of parallelism occurs within the Spark job for each entity where we parallelise the computation of the influence probabilities

[5]https://hbase.apache.org/

[6]http://spark.apache.org/

[7]N.b. the maintenance of the interactions between users records both the source and target of the interaction, hence we can retrieve directed interactions both ways - i.e. $u \rightarrow v \wedge v \rightarrow u$.

TABLE I
TO DO

			Probability Setting
		Static	Discrete Time ($t_v \in [t_u - \tau_{u,v}, t_u)$)
Model	p_u^E (Actions Propagation-based)	0.722	
	p_u^I (Interactions-based)		
	p_u^C (Community-homophily)		

between two users - although this parallelism occurs *within* each iteration through the time-ordered set of entity posts. The Java implementation of this code can be found in the github repository,[8] including the functions for building the HBase tables, deriving the entity-propagation counts (E_{v2u}) and the test algorithm.

2) Results: Table I presents the results from our experiments of the various models and probability settings

a) Per-Entity Results: -Plot the distribution of per-entity roc values for the above models

VII. Discussion and Future Work

-Defend the adoption-exposure claim of mode at 0, and explain how validation of this claim is planned in our future work using regexes over the entirety of Reddit to speed up computation.

FW1: -Compartment Models of community-wise diffusion

FW2: -Vulnerability windows and colinear diffusion patterns

References

[1] J. Baumgartner, "Complete public Reddit comments corpus," 2015.

[2] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.

[3] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory Search and Topic Summarization for Twitter," in *Proc. ICWSM*. AAAI, 2010.

[4] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, "Analysis of named entity recognition and linking for tweets," *Information Processing & Management*, vol. 51, no. 2, pp. 32–49, 2015.

[5] A. Ritter, S. Clark, O. Etzioni *et al.*, "Named entity recognition in tweets: an experimental study," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 2011, pp. 1524–1534.

[6] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh Conference on Natural Language Learning*. ACL, 2003, pp. 142–147.

[7] T. Baldwin, Y.-B. Kim, M. C. de Marneffe, A. Ritter, B. Han, and W. Xu, "Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition," *Proc. W-NUT (ACL-IJCNLP)*, pp. 126–135, 2015.

[8] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[9] J. Turian, L. Ratinov, Y. Bengio, and D. Roth, "A preliminary evaluation of word representations for named-entity recognition," in *NIPS Workshop on Grammar Induction, Representation of Language and Language Learning*, 2009, pp. 1–8.

[10] L. Derczynski and S. Chester, "Generalised Brown Clustering and Roll-Up Feature Generation," in *Proceedings of the 30th conference of the Association for Advancement of Artificial Intelligence*, 2016.

[8] https://github.com/mattroweshow/NER-Diff-Paper

[11] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2009, pp. 147–155.

[12] L. Derczynski, I. Augenstein, and K. Bontcheva, "USFD: Twitter NER with Drift Compensation and Linked Data," *Proc. W-NUT (ACL-IJCNLP)*, pp. 48–53, 2015.

[13] L. Derczynski, S. Chester, and K. S. Bøgh, "Tune Your Brown Clustering, Please," in *Proceedings of the conference on Recent Advances in Natural Language Processing (RANLP)*, 2015.

[14] S. Chester and L. Derczynski, "generalised-brown: Source code for AAAI 2016 paper," Nov. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.33758

[15] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[16] J. Leskovec, M. McGlohon, C. Faloutsos, N. S. Glance, and M. Hurst, "Patterns of cascading behavior in large blog graphs." in *SDM*, vol. 7. SIAM, 2007, pp. 551–556.

[17] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "An improved algorithm for matching large graphs," in *3rd IAPR-TC15 workshop on graph-based representations in pattern recognition*. Citeseer, 2001, pp. 149–159.

[18] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 241–250.