

# Don't Let Me Be #Misunderstood: Linguistically Motivated Algorithm for Predicting the Popularity of Textual Memes

Oren Tsur<sup>†‡</sup> Ari Rappoport<sup>§</sup>

orentsur@seas.harvard.edu arir@cs.huji.ac.il

<sup>†</sup>School of Engineering and Applied Sciences, Harvard University

<sup>‡</sup>Lazer Laboratory, Northeastern University

<sup>§</sup>School of Computer Science and Engineering, The Hebrew University

## Abstract

Prediction of the popularity of online textual snippets gained much attention in recent years. In this paper we investigate some of the factors that contribute to popularity of specific phrases such as Twitter hashtags. We define a new prediction task and propose a linguistically motivated algorithm for accurate prediction of hashtag popularity. Our prediction algorithm successfully models the interplay between various constraints such as the length restriction, typing effort and ease of comprehension. Controlling for network structure and social aspects we get a glimpse into the processes that shape the way we produce language and coin new words. In order to learn the interactions between the constraints we cast the problem as a ranking task. We adapt Gradient Boosted Trees for learning ranking functions in order to predict the hashtags/neologisms to be accepted. Our results outperform several baseline algorithms including SVM-rank, while maintaining higher interpretability, thus our model's prediction power can be used for better crafting of future hashtags.

## Introduction

A growing body of work researches the factors that contribute to the popularity, propagation and trajectory of cascades of shared pieces of information (memes) in online communities. Mosts of these works investigate these processes as a function of network factors and trajectories of time series of the propagation in early stage of the meme's life cycle. Some works take the meme's broad topical context and some content structure into account (Leskovec, Backstrom, and Kleinberg 2009; Berger and Milkman 2012; Tsur and Rappoport 2012; Ma, Sun, and Cong 2013; Cheng et al. 2014). In this work we present a new task – predicting the most popular meme in its immediate context – semantically interchangeable set of textual alternatives.

We hypothesize that a “winner” has some inherent advantages over competing alternatives. Inspired by the linguistic Optimality Theory paradigm (Prince and Smolensky 1997) we propose a linguistically motivated algorithm for this prediction task. We refer to memes as neologisms and try to

learn the weights of various constraints that play role in the emergence of such “neologisms” in this specific domain.

## Neologisms

Neologisms are newly coined words or phrases commonly used by a discourse community. Some neologisms that are already lexicalized and absorbed into the standard language are ‘brunch’, ‘cyberspace’, ‘laser’, ‘mcjob’, ‘internet’ and ‘meme’<sup>1</sup>. While neologisms pose many problems to Natural Language Processing (NLP) systems (Cook 2010), the formation and acceptance patterns of neologisms are of a major interest from pure (socio/psycho-) linguistic perspective (Brinton and Traugott 2005; Deutscher 2006).

Typically, neologism are studied under a qualitative paradigm in which scholars examine the etymology of selected words (Algeo 1977; 1980). Quantitative approaches to language-change are gaining popularity as resources like Google Books are becoming available (Lieberman et al. 2007; Michel et al. 2011; Petersen et al. 2012; Goldberg and Orwant 2013). These works examine the language change in hundreds of years perspective. The evolution of language, however, is continuous and takes place in much shorter time frames (Deutscher 2006). The massive data stream available in social networks is giving rise to the possibility of studying language change in real time.

Neologisms, similarly to the communication process in general, embody inherent tension between two main forces: the speaker's goal to be well understood and the ‘principle of least effort’, which argues that a word's frequency is inversely proportional to its length (Zipf 1949), coinciding with the effective communication theory (Shannon et al. 1949). This approach to language was criticized (e.g. Chomsky (1976)). In effect, in the past decades the effective communication framework was not widely applied to natural language<sup>2</sup>. Recently, however, this framework regained interest due to the growing availability of large textual corpora. It has been demonstrated that content is a good predictor for word length (Piantadosi, Tily, and Gibson 2011) and that speakers choose shorter words in predictive con-

<sup>1</sup>The term *meme* was actually coined back in the '70 by Richard Dawkins, meaning an idea that spreads itself with some variations.

<sup>2</sup>With some notable exceptions in *speech* processing e.g. (Lindblom 1990).

texts (Mahowald et al. 2013).

## Hashtags as Neologisms

Twitter hashtags can be viewed as new phrases coined by the user community as the need arises. Given the need for a new phrase, a number of optional hashtags can be coined (e.g. *#bindersFullOfWomen*<sup>3</sup>, *#bindersfullofwomen*<sup>4</sup>, *#romneysBinders*, etc.), some of which gain popularity while others remain marginal.

In this paper we examine the popularity of hashtags in light of linguistic, cognitive and domain constraints. By modeling the interplay between various constraints we get a glimpse into the processes that shape the way we coin and adopt new words – to what extent the principle of least effort holds and under what circumstances it is violated.

We define a new prediction task and train an algorithm to predict the preferential ranking of competing words describing the same concept. Specifically, we look at small sets of competing ‘interchangeable’ hashtags, predicting which of the hashtags in each set will be used more frequently.

We note that while hashtags are exclusive to the online language, their role in the Twitterverse (and other online services) has exceeded beyond the original purpose of tagging and indexing and they provide the user community a fertile ground for creative puns and playful use of language.

## Algorithmic Approach

In order to learn the ranking model we adapt the GBrank framework proposed by (Zheng et al. 2007) to our task. In this framework, a Gradient Boosted Trees (GBT) algorithm, typically used to fit a regression function, is modified to enable ranking of preference pairs. The conversion of learning a regression function to learning a ranking function is achieved via a modified update mechanism in the learning process. Using the GBrank framework, we achieve superior results in the prediction task. The algorithm efficiently produces nonlinear models in various levels of complexity while maintaining relative interpretability, that can be easily translated into linguistic insights such as the contexts in which efficiency is preferred and what language conventions are more prone of violation.

**Contribution** The main contributions of this paper are: (1) We introduce a new prediction task in which popularity of hashtags is predicted in comparison to the immediate context, (2) We propose a linguistically motivated ranking algorithm and demonstrate its success in the prediction task, (3) The proposed algorithm produces models that are easy to interpret, comparing to other models thus we gain a glimpse into the cognitive linguistic process of language generation. These insights can be applied to improve crafting of new

<sup>3</sup>The phrase ‘binders full of women’ was used by Mitt Romney in the second presidential debate of 2012 in his response to a question about pay equity. The phrase quickly turned viral.

<sup>4</sup>We view different capitalizations of the same sequence of characters as different hashtags as they require different typing efforts (from the writer) and different cognitive efforts (from the reader).

hashtags. (4) We offer an extensive discussion on some social and linguistic phenomena that interfere with the learning model.

## Ranking Algorithm

There are two requirements we want our algorithmic framework to meet: (1) The model should allow accurate prediction of the speakers preferences. and (2) The model should be as interpretable as possible in order to gain some (psycho/cognitive)-linguistic insights about the processes governing the production of the text.

In order to meet these two criteria, we train a model based on linguistic and domain-constraint features. The model should allow us to learn the relations between various constraints. In the light of these requirements we formulate the problem as a ranking task.

**Order and preferences:** Given a set of hashtags  $HT$  and a corresponding set of target values  $Y$ , we define a preference order  $ht_i \succ ht_j$  if  $y_i > y_j$ . We denote the preference pair as an ordered set  $\langle ht_i, ht_j \rangle$ . Although the values in  $Y$  can introduce a preference order over all pairs in  $HT \times HT$ , only a partial set of pairs is considered, namely only pairs in which both elements belong to the same subset. Given  $C$ , a partition of  $HT$  to subsets of vectors (hashtags) such that  $\cup_k c^k = C$ , we define the set of relevant pairs as:

$$S = \{ \langle ht_i^c, ht_j^c \rangle | i \neq j, ht_i^c \succ ht_j^c, c \in C \} \quad (1)$$

**Notation:** For the ease of reading, where context allows, we use  $ht_i$  to represent both the  $i$ 'th hashtag in a set and its corresponding vector. We also simplify the notation when possible, omitting the  $c$  from notations like  $ht_i^c$ , given that the preference relation is only meaningful for hashtags from the same set  $c$ .

**Ranking Function:** We are interested in learning a function  $h \in H$  such that  $h$  projects  $h(ht_i^c) \geq h(ht_j^c)$  for any given preference pair  $ht_i^c \succ ht_j^c$  (reads as ‘in pair  $\langle ht_i, ht_j \rangle$  from the same cluster  $c$ ,  $ht_i$  is preferred over  $ht_j$ ’).

Given a set of preferences  $S$  as defined in Equation 1, the objective loss function to minimize is defined as:

$$L(h) = \frac{1}{2} \sum_{\langle ht_i, ht_j \rangle \in S} (\max\{0, h(ht_j) - h(ht_i)\})^2 \quad (2)$$

meaning there is a cost only for explicit violation of the true order<sup>5</sup>.

In the classic boosting framework, a set of base (weak) learners can be combined to create a single stronger learner (Schapire 1990). Zheng et al. outline a regression framework for learning of ranking functions using relative relevance judgments and click through data (Zheng et al. 2007). They present GBrank – an algorithm based on gradient boosting that is used for reranking the web pages returned for a search engine query.

<sup>5</sup>Theoretically, an artificially optimal solution could be achieved by fixing  $h(ht) = b$  for any constant  $b$ , however, this solution is impossible given the proposed algorithm.

Our task can be viewed in a similar way: given a concept, impose a preference order on the possible hashtags that may be used in lieu of that concept. In this paper we adapt and modify the GBrank framework to suit the task at hand.

GBrank iterates between two phases: (1) fitting a regression function, and (2) rescaling the regression target values in order to minimize the ranking loss function. Each iteration adds a new base learner thus the combination of the base learners is expected to boost performance. Rescaling is performed in a gradient descent manner.

It is important to note that each phase of the algorithm optimizes a different objective function. Phase one fits a regression function by minimizing the loss function. This optimization can actually work against our goal of learning ordered pairs. The gradient rescaling in phase two is expected to calibrate the target values so that the new regression function keeps the given order.

Given a preference pair  $\langle ht_i, ht_j \rangle$ , the corresponding gradients of  $L(h)$  (Equation 2) are:  $\max\{0, ht_j - ht_i\}$  and  $\max\{0, ht_i - ht_j\}$ , hence the gradient is different than zero only if there is an explicit violation of the order of the preference pair.

In order to use these gradients in the ranking scheme we add (or subtract) some fixed values  $\tau$  to the regression target values. Each consecutive base learner is a regression tree that maximizes the fit with the new target values.

Different base learners can be assigned different weights in order to introduce regularization and prevent overfitting.

## Modified GBrank algorithm

Our modified algorithm is presented in Table 1. The reader should note the difference between sets. While  $T, T^+$  and  $T^-$  that are used for learning a regression function ( $g_k$ ) contain pairs, each pair consists of a vector and its (adjusted) currently predicted value;  $S, S^+$  and  $S^-$  are sets of pairs of vectors and are used in order to adjust the currently predicted values to reflect the desired ranking.

In our experiments we use  $\alpha = 1/2$  and  $\beta = 1$ , thus the training set used in the construction of each new tree is composed of the hashtags from half of the correctly ranked pairs and hashtags from half of the wrongly ranked pairs.

Trees construction is optimized (greedily) by adding decision nodes that maximize the information gain. Decision cutoff is decided according to a weighted average of samples in the sets divided by the feature in the relevant node.

The major modification introduced to GBrank is that in stage 4 we use a weighted majority vote while (Zheng et al. 2007) use the following linear combination of the  $i = 1, \dots, k$  regression trees  $g_i(ht)$  as a ‘pseudo’ ranking function:

$$h_k(ht) = \frac{k \cdot h_{k-1}(ht) + \eta_k \cdot g_k(ht)}{k + 1}$$

Experimenting with this pseudo ranking function, the learner indeed managed to minimize the regression loss function, however as model complexity grows (more trees/levels) the results get worse due to over correction by subsequent trees trying to compensate for the growing target values (stage 3). Convergence to sub optimal solution is finally achieved due to the shrinkage factor  $\eta_k$ , practically

**Input:** A set of vectors and their corresponding target values  $HT : Y$ ; a set of preferences  $S$  defined in Equation 1.

**Output:** A function  $Rank : HT^c \times HT^c \rightarrow \{1, -1\}$ , where  $Rank(ht, ht') = 1$  iff  $ht \succ ht'$ .

**Initialize:** ‘Guess’ initial target values for  $g_0$ , e.g.  $g_0(ht) = 0$  for each  $ht$ .

**For**  $k = 1, 2, 3, \dots$

1.  $\hat{Y}^{k-1} = \langle \hat{y}_1^{k-1}, \hat{y}_2^{k-1}, \hat{y}_3^{k-1}, \dots \rangle$   
where,  $\hat{y}_i^{k-1} = g_{k-1}(ht_i)$

2. Divide  $S$  to  $S^+$  and  $S^-$  as follows:  
 $S^+ = \{(ht_i, ht_j) \in S | \hat{y}_i^{k-1} - \hat{y}_j^{k-1} > \beta\}$   
 $S^- = \{(ht_i, ht_j) \in S | \hat{y}_j^{k-1} - \hat{y}_i^{k-1} > \beta\}$   
where  $\beta$  is some non-negative value.

3. Use Gradient Boosted Trees to fit a regression function  $g_k(ht)$  on the following training set  
 $T = \text{sample}(T^+, \alpha) \cup \text{sample}(T^-, \alpha)$

where,

$T^+ = \{(ht_i, \hat{y}_i^{k-1}), (ht_j, \hat{y}_j^{k-1}) | (ht_i, ht_j) \in S^+\}$

$T^- = \{(ht_i, \hat{y}_j^{k-1} + \gamma), (ht_j, \hat{y}_i^{k-1} - \gamma) | (ht_i, ht_j) \in S^-\}$

$\alpha$  is the subsampling factor

4. Update the ensemble  $Rank_k(ht, ht')$  with  $r_k$ :

$$Rank_k(ht, ht') = \begin{cases} 1 & \sum_{i=1}^k \eta_k r_i(ht, ht') > 0 \\ -1 & \sum_{i=1}^k \eta_k r_i(ht, ht') < 0 \\ \text{rand}(-1,1) & \text{otherwise} \end{cases}$$

where

$$r_k(ht, ht') = \begin{cases} 1 & g_k(ht) > g_k(ht') \\ -1 & g_k(ht) < g_k(ht') \\ 0 & \text{otherwise} \end{cases}$$

Table 1: Modified GB rank algorithm.

preventing further changes in the prediction. We therefore propose the weighted majority function  $Rank(\cdot)$  in which each tree provides its own preference prediction as outlined above. In our experiments the modified GBrank achieved improvement of about 40% over the original GBrank proposed by Zheng et al. Another modification we introduced is the subsampling in stage 3. Subsampling is used as it is reported to improve results significantly in many settings (Friedman 2002). The training set of each new base learner is combined from a sampled subset of the wrongly ordered pairs and a sampled subset of the the original training set (with the adjusted target values).

## Data

### Twitter Corpus and statistics

A Twitter posting is called a *tweet*. A tweet is restricted to 140 characters in length. This length constraint makes characters ‘expensive’, hence tweets present an informal language and introduce many abbreviations. Twitter allows the use of two meta characters: ‘@’ marking a user name (e.g. @BarackObama), and ‘#’ marking a hashtag: a sequence of non whitespace characters preceded by the hash character

(e.g. #healthCareReform).

Hashtags are used extensively and are adopted organically as part of the dynamic communication process. The use of hashtags is a popular way for a user to provide his readers with some context<sup>6</sup>, an important function due to the length constraint. For example the hashtag #savethenhs, reads as ‘save the national health service’, gives the context relevant to the tweet “Speaker refers to #Lanseys ‘abysmal ignorance’ as demonstrated on alcohol strategy; this SoS #nhs #savethenhs #healthbill”. It is important to note that from a functional perspective (indexing/search) capitalization does not play any role in the hashtag format, therefore #NHS, #nhs and #nhS are equivalent, although very different to the human eye.

### Corpus statistics

Our basic corpus consists of over 417 million tweets from Twitter, collected from June 2009 through December 2009. This corpus is a sample of approximately 15% of the Twitter stream in this six month’ period. Over three million unique hashtags were observed in our data in over 49 million occurrences, an average of 0.11 hashtags per tweet. The hashtag frequency presents a long tail distribution where the 1000 most frequent hashtags (0.003% of the unique hashtags) cover 43% of hashtag occurrences. 67% of the unique hashtags appear only once.

We opted for this 2009 corpus in order to control for ‘formal’ or ‘institutionalized’ hashtags we further discuss in the Institutionalization Subsection in the Results Section. In 2009 Twitter was popular enough to serve as a large scale active arena for social interaction, presenting complex patterns of information diffusion while still not “abused” by mainstream institutions and campaigns.

### Competing interchangeable hashtags

As described in the introduction, we are interested in predicting preference order within sets of interchangeable hashtags. Set construction is problematic for two main reasons – fuzzy similarity and community disparity.

**Fuzzy similarity** Finding or defining small sets of similarly used hashtags is a great challenge and the fine granularity of clusters required in our task is beyond state of the art clustering algorithms for sparse data (Tsur, Littman, and Rappoport 2013). For example, given the following hashtags: #iranElection, #freeIran, #greenRevolution, #IranElection, #usElection, #iranelection, #followFriday, it is clear that some of these hashtags should be classified under the general class ‘Iran’ or more specifically ‘Iran election’, however it is hard to determine whether #freeIran is interchangeable with #iranElection and/or with #greenRevolution<sup>7</sup> or not.

<sup>6</sup>The exact functionality of a hashtag is defined by the practice of the community that uses it. The most frequent uses of hashtags are as a topic marker, as a **bold** typeface intended to draw attention to an important concept and a marker for the agenda of the tweeter.

<sup>7</sup>In fact, these hashtags are rather similar, being promoted by similar users in similar contexts. #freeIran and #iranElection are used in tweets related to the allegedly corrupted election in Iran and ‘green’ (as in #greenRevolution) is the color of the Iranian opposition activists and became the color of the demonstrations against

Rank	Iran election	Gaga VMAs	jobs
1	#iranelection	#GagaVMAs	#jobs
2	#IranElection	#GaGaVMAs	#Jobs
3	#Irenelection	#GagaVMAS	#job*
4	#iranElection	#gagavmas	#Job*
5	#iranelections*	–	#JOBS
6	#IRANELECTION	–	#JOB*
7	#IranElections*	–	–
8	#iranelect*	–	–
9	#IRANElection	–	–
10	#IRanElection	–	–

Table 2: Three ordered sets (1 is the most popular). Iran Election refers to the allegedly corrupted elections of 2009, Gaga VMAs refers to the Lady Gaga’s nomination for MTV’s Video Music Awards, and jobs refers to the job market after the 2008 crisis. Asterisks indicate hashtags with shared stems that are added to the conservative clusters.

**Community disparity** Although interchangeable, different hashtags are sometimes used within different subcommunities therefore their frequency is highly affected by the graph topology and the influence structure of the members of the sub-communities. Accounting for the complexity of the network, social status, influence and the diffusion processes is beyond the scope of this paper and partially addressed in Danescu-Niculescu-Mizil et al. (2011; 2013) and Tsur and Rappoport (2012). In this study we focus on linguistic aspects, thus non-linguistic effects must be controlled for.

Due to the two reasons mentioned above we rather use two relatively conservative formulations of sets. In the first setting (CAPS) we cluster together hashtags that only differ in capitalization. In the second setting (STEM) we cluster together hashtags that share similar stems of the words composing them (e.g. grammatical suffixes like ‘ed’, ‘ing’, etc.).

Table 2 presents a few examples of such clusters. The ranking of the hashtags within a group is based on the normalized frequency of the hashtag in the corpus. Normalization is required since the daily sample size released by Twitter was changed from time to time and since the overall volume of Twitter messages grew considerably during the six months the data was collected.

In order to have a meaningful ranking, we only looked at hashtags that appeared more than a hundred times assuming small counts are prone to bias and many times indicate unintentional typos. Filtering out the most infrequent hashtags, our experimental data contains a total of 15046 hashtags. In the conservative CAPS setting, sets size are of 2 – 11 tags, with the average of 2.27 hashtags per set and a standard deviation of 0.6.

### Model features

In order to learn the model we represent each hashtag as a feature vector. Since our motivation is to model the dynamics between the various constraints we use features that may play a role in the language production process.

the results of the official election.

Our simplified hypothesis is that a speaker wishes to be well understood by her audience while minimizing the effort of producing the speech act<sup>8</sup>. We consider four major generic forces in our model: (1) production effort, (2) comprehension effort, (3) linguistic habits and conventions (possibly shaped by the production and comprehension efforts), and (4) Domain-incurred constraints, in our case the tweet's length restriction of 140 characters. These four forces are introduced to our model as families of features in a feature vector.

We expect these feature to interact in a non-linear way, for example, consider the following four hashtags (1) *#savethenhs*, (2) *#savethenationalhealthservice*, (3) *#saveTheNHS* and (4) *#saveTheNationalHealthService*. (1) and (2) require less typing effort than the (3) and (4) (respectively), while the latter two are easier to comprehend since tokenization is made easier by the capitalization of word initials. Assuming that the community is already familiar with the abbreviation (NHS), it is clear that not only that typing *#saveTheNHS* is more efficient than typing *#saveTheNationalHealthService*, it is also easier to read and comprehend. Moreover, we expect (3) to be preferred by the users for its brevity in the light of the length constraint imposed by Twitter (although increased brevity, e.g. *#stnhs* will probably fall short for its high comprehension effort).

In the remainder of this section we list the features and provide a brief explanation for the intuition behind using these specific features.

**Number of characters** The length of a hashtag plays an important role due to the 140 characters length constraint as well as to its effect on correct encoding of the concept it represents, ease of typing and ease of reading and comprehension.

**Number of words** Hashtags can be combined from several words. The tokenization of the hashtag into distinct words is a demanding cognitive task on the reader's side.

**Number of shift keystrokes** the number of strokes on the shift key represents the estimated production effort in the textual domain. The actual production effort can vary according to the device used (e.g. a full size keyboard vs. mobile device) and the application interface. Some noise is incurred by retweeting other messages (see below). While the use of special characters such as the underscore (*\_*) or capital letters requires more typing effort, the use of capital and special characters can improve comprehension, compare *#savethenationalhealthservices* to *#saveTheNationalHealthService* and *#save\_the\_national\_health\_services*.

**Retweet rate** A 'retweet' is a tweet that is shared ("forwarded", promoted) by another user. Retweeting is easy as the user does not have to retype the whole message but only

<sup>8</sup>Obviously this hypothesis is simplified and there may be many utterances that do not directly support this hypothesis, e.g. sarcasm, politeness and other pragmatic uses of language. For some discussion see (Grice 1975).

click the 'retweet' key - minimizing language production effort even in cases where the original typing is hard<sup>9</sup>. While the retweet rate is not a linguistic feature per se, we expect it to contribute to the model, interacting with other features. As retweet rate is a global feature, we verified it is not correlated with the hashtag frequency and cannot be used as a single feature for accurate prediction of the ranking (correlation is 0.03).

**Named Entities** In standard English named entities are capitalized. The user may or may not "import" this writing convention into Twitter even though it require some extra effort.

**Proper name** In Standard English proper names are capitalized therefore users may "import" this habit to Twitter even though it requires extra key strokes and has no explicit contribution (e.g. *#obama* vs. *#Obama*).

**Location** In standard English locations are typically capitalized (Washington). Many locations are combined from more than one word (New York, New York City) and they are often abbreviated (UK, NY, NY City, NYC).

**Acronyms** Acronyms are typically written with capital letters for enhanced comprehension.

**All caps** All caps can be used in order to mark an acronym or stress out part of the sentence/hashtag. From text production perspective, it is easier to capitalize a sequence of letters than capitalize only some of the letters (leaving a finger on the Shift key or using the Caps Lock).

Named entities (including Location and Proper Name) and acronym are binary feature with the value 1 iff (part of) the hashtag belongs to the respective class. All-caps is 1 iff all hashtag letters are capitalized.

The following features are only relevant to the STEMS setting as they differentiate between hashtags with different characters.

**Levenshtein distance** The Levenshtein distance between the hashtag and its stemmed version. For example, the stemmed version of the hashtag *#NetworkingWitches* is *network,witch* and the Levenshtein distance is 5 (not counting capitalization differences). This feature is expected to represent the distance of the hashtags in the similarity set from a canonical form of the set's concept.

**Suffixes** This group of features contains three different binary features, indicating whether the hashtag contains either *ed*, the *s* or the *ing* suffixes. This set of features is designed to capture the use of grammatical suffixes and the user's tendency to omit/use them with regards to other features.

<sup>9</sup>Depending on the application interface used, a retweet button might not be available, in that case the user copy and paste the original message, adding the token 'RT' (or 'RT @[user name]') at the beginning of the message, e.g. RT @support Adding a mobile number to your account can help you recover your password down the road is a retweet of the original message posted by Twitter's @support.

$\delta$	1	10	50	100	1000
pairs in CAPS	10575	9487	6753	5266	1324
pairs in STEMS	22762	20201	14278	10816	2445

Table 3: Number of remaining preference pairs in each setting, given various values of  $\delta$  as a preference threshold.

## Experimental Framework

### Preference pairs

The Corpus Statistics Section presents general corpus statistics. In this section we provide some more details relevant to the specific experimental framework. Initially, 6636 clusters were extracted from our corpus, containing a total of 15046 hashtags that constitute 10757 preference pairs.

‘Preference’ is defined by the normalized frequency of the hashtags in the corpus, thus for a pair  $\{ht_i, ht_j\}$  we say that  $ht_i \succ ht_j$  (or  $\langle ht_i, ht_j \rangle$ ) if  $count(ht_i) > count(ht_j)$  and  $ht_i \prec ht_j$  ( $\langle ht_j, ht_i \rangle$ ) if  $count(ht_i) < count(ht_j)$ . However, since these preferences are decided by corpus statistics (usage counts), some of the preferences can be attributed to chance or to other non-linguistic factors such as retweeting. In order to prevent the randomness incurred by counts, we introduce  $\delta$ : a preference-strictness threshold, thus  $S$ , the set of preference pairs is slightly altered:

$$S = \{\langle ht_i, ht_j \rangle | d(ht_i, ht_j) \geq \delta\} \quad (3)$$

where  $d(ht_i, ht_j)$  is defined by:

$$d(ht_i, ht_j) = count(ht_i) - count(ht_j)$$

Table 3 gives the number of pairs remaining after applying various values of  $\delta$  in formula 3.

### Baseline algorithms

**Least-effort baseline** As this work is inspired by the ‘least-effort’ paradigm that is even more prevalent in the informal domain of Twitter, it is straight forward to use a ‘least effort’ algorithm as the baseline. Given a pair  $(ht_i, ht_j)$  the baseline predicts the shorter hashtag to be preferred. In the conservative context of this work *shorter* means the one requiring less typing effort where each key stroke counts. For example, the hashtag *#freeIran* will be preferred over *#FreeIran* due to the extra press on the Shift key producing a capital *F*. Similarly, *#truth* is expected to be preferred over *#Truth* and *#Truth* over *#TRUTH*.

**SVM-rank** A modification of the SVM classifier to use ordinal regression, indicating a rank instead of a nominal class (Joachims 2006)<sup>10</sup>.

<sup>10</sup>(Caruana and Niculescu-Mizil 2006) show that in some settings boosted trees learners outperform SVM learners on binary classification tasks. In this paper we use modified algorithm for a ranking prediction task.

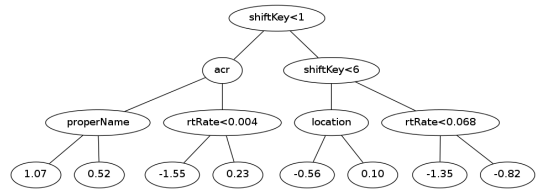


Figure 1: An example of a tree created by GBrank. The values at the leaves are the regression prediction values. The labels at the decision nodes indicate the feature decided upon. Nodes without explicit condition means an existence condition, i.e. in a *properName* node the *existence* of a proper name in the hashtag is checked (e.g. *#obamaCare*). Satisfying the condition in nodes of the form *feature* < *n* leads to the left branch, satisfying the condition in nodes of the form < *label* > leads to the right branch.

## Results

Table 4 presents the error rates of the GBrank, the SVM-rank<sup>11</sup> and the least-effort baseline in both STEMS and CAPS settings and with varying strictness of preference ( $\delta$  values).

While the error rate of the SVMrank is slightly lower than the least-effort baseline (except for STEMS with  $\delta = 1$ ), GBrank achieves significantly lower error rates in all settings. As expected, increased strictness in the preference definition (as defined by Formula 3) improves prediction results, supporting the intuition that the ‘real’ preferences defined by the frequency of mentions in social networks is prone to some noise that should be accounted for.

Comparing the STEMS and the CAPS settings, it seems that the STEMS setting poses a bigger challenge to all algorithms. Examining the data we verified that while the sets in conservative CAPS setting are relatively noise-free (there is some noise between sets), the similarity sets in the STEMS setting bear some noise as hashtags with different suffixes (e.g. *job* and *[Steve]jobs*), do not always conform to the exact same concept. In order to provide a cleaner discussion the rest of the paper refers mostly to the results of the CAPS setting.

It is well worth noting that although the CAPS setting seems restrictive, not only it provides a much cleaner look on the writing preferences, it may also reflect a fundamental linguistic phenomena as it somewhat resembles Optimality Theory approaches for modeling changes in the placement of the stressed syllable in a word and in the placement of expletive infixations (Hammond 1996).

One of the main motivation for our work is to learn how the interplay between different constraints affects hashtags crafting and acceptance. GBrank produces nonlinear, interpretable, models that allow us to ‘extract’ the rules that govern this process. An example of such rules is provided by the trees in Figure 1. Note that the role of a tree  $T_k$  in the model is to compensate for errors and cases that were not handled

<sup>11</sup>The SVMrank is used with a linear kernel function. All reported results are for 5-fold cross validation. Using polynomial or rbf kernel did not achieve convergence.

Setting	Algorithm	$\delta = 1$	$\delta = 50$	$\delta = 100$
STEMS	least effort	0.383	0.34	0.314
	SVMrank	0.39	0.329	0.304
	GBrank	0.266	0.146	0.16
CAPS	least effort	0.32	0.25	0.214
	SVMrank	0.31	0.24	0.2
	GBrank	0.16	0.12	0.07

Table 4: Error rates of the least-effort baseline, the SVMrank and the GBrank in various levels of preference strictness ( $\delta$  values). GBrank achieves the lowest error rate in all settings. GBrank setting: number of trees: 7, tree levels: 3,  $\eta_k = 1$  and  $\tau = 1$ . Error rates are averaged for 5-fold cross validation.

Levels / Trees	1	3	5	7	10	15
1	0.26	0.22	0.2	0.19	0.18	0.16
2	0.25	0.19	0.175	0.14	0.1	0.1
3	0.24	0.15	0.12	0.12	0.1	0.09
4	0.22	0.13	0.12	0.1	0.08	0.07
5	0.21	0.13	0.1	0.8	0.08	0.07

Table 5: Error rates of models with different complexities according to the number of trees (columns) and the maximal depth of the trees (rows). Error rates are averaged for 5-fold cross validation for the CAPS setting with  $\delta = 50$ ,  $\eta_k = 1$  and  $\tau = 1$ .

by trees  $T_{0,\dots,k-1}$ . In most cases, some of the trees will be idle, predicting ‘no preference’ since no decision node differentiates between the candidates in the preference pair. In case the ensemble is idle, preference is decided randomly. In the scenario of 7 trees, 3 levels and  $\delta = 50$ , the reported result was achieved with an average majority of 2.43 (of the seven trees) and standard deviation of 1.47. The idleness of some of the trees for a candidate preference pair, allows easier interpretability while allowing models with higher complexity combining a large number of trees.

### Confidence and idleness

The average majority differences presented in Table 6 reflect the ‘confidence’ of a trained ranker. A maximal confidence score would be the number of trees in the model, indicating that all trees agree on all pairs in all folds. Maximal score is never achieved in practice since the different trees may capture different linguistic phenomena and predict preference accordingly. Some trees may also be idle for a given pair of hashtags as the features captured in these trees might not differentiate between the hashtags in the given pair.

Table 6 shows the average difference in the majority vote (stage 4 of the GBrank algorithm in Table 1). An average difference of 3.03 (7 trees, 3 levels,  $\delta = 50$ , CAPS setting) means that preference prediction was made by an average majority of 3.03 trees. The interpretation of this difference can indicate ‘decisiveness’ on the range between two extreme cases: (a) four out of the seven trees are idle in the decision making (possibly different four for different pairs), and (b) prediction is based on five trees ‘preferring’ a specific order and only two trees predicting the opposite, resulting in a difference of three in the majority vote. The average

#trees	1	2	3	4	5	6	7
STEMS	1	1.2	1.29	1.42	1.52	1.56	1.63
CAPS	1	1.3	1.78	2.07	2.37	2.72	3.03

Table 6: Average difference in the majority vote of trees (Stage 4 in the algorithm in Figure 1) for 5-fold cross validation  $\delta = 50$  and  $level = 3$ .

difference of 3.03 achieved by our algorithm indicates a relatively strong confidence even under interpretation (b).

It is notable that as the model is boosted by more trees the confidence increases. It is also worth noting that the confidence values in the CAPS setting are significantly higher than the respective values in the STEMS setting (for models with more than a single tree). This clear trend reflects the fact that the CAPS setting is much cleaner than the noisy STEMS setting – similarly to the interpretation of the results presented in the previous subsection (and in Table 4).

### Manipulating model complexity

A Gradient Boosted Trees method allows experimenting with models of various complexities. The complexity of a model is subject to three factors: the number of attributes spanning the vector space, the number of trees in the model and the maximal depth of the trees. More complex modes can capture subtler phenomena, under the risk of overfitting and/or loss of interpretability.

Figure 1 presents the first tree learned by the algorithm, demonstrating the interplay between the various constraints. Table 5 presents results for various complexities. While increased complexity seems to improve results, complex models (e.g. 15 trees of  $depth = 5$ ) are hardly interpretable. Learning a single tree of depth 7 harms the results (0.22 error rate), clearly indicating overfitting. Growing additional trees with subsampling prevents overfitting while harming interpretability.

We note that interpretability is possible even in the case of multiple (shallow) trees, as many trees are idle regarding most hashtags (though different trees are idle for different hashtags, see some numbers above).

### Discussion

Brevity is the single most significant signal, considering that the ‘least effort’ baseline has an error rate of 25% (CAPS,  $\delta = 50$ ). User efficiency is indeed justifiable in many cases: 40% of the hashtags are single-word hashtags that are not named-entities. As expected, the models learned by our modified GBrank effectively model the interaction between nonlinear constraints (such as the number of words, characters, abbreviations etc.). Some less trivial, yet correct, predictions made by our algorithm are:  $\#cot \succ \#TCOT \succ \#Tcot$  (acronym of ‘top conservative’, used by American republicans),  $\#TV \succ \#Tv$ ,  $\#BadRomanceiTunes \succ \#badromanceitunes$  and  $\#GagaVMAs \succ \#GaGaVMAs \succ \#gagavmas$  (compare to order at Table 2).

Two interesting cases of correct predictions are  $\#obama \succ \#Obama$ ,  $\#London \succ \#london$ . Both preference pairs are single-word named-entities. The correct ranking suggests

that named entities should be broken down to types and that some types (of limited length) are more prone to “violate” the capitalization of proper named rather than those of location entities (depending on the number of characters).

In the remainder of the section we discuss three of the possible causes for error (beyond the noise naturally occurring with real world data): ambiguity, canonization and institutionalization.

### Ambiguity

Our data collection contains many ambiguous words, mainly single-word named-entities. Two examples of such cases are the hashtags *#jobs* (see Table 2) and *#dolphins*. While ‘jobs’ can refer to the job market, thus capitalization does not contribute to comprehension, it can also refer to Apple’s former CEO Steve Jobs, in which case capitalization may or may not occur. Similarly, *#dolphins* can refer to the ecologist battle cry “save the *#dolphins*” or refer to the Miami Dolphins NFL team. Disambiguation of hashtags is beyond the scope of this paper and will be addressed in future work.

### Canonization

Some hashtags gain such popularity that they become part of the standard (Twitter) language. In these cases, the hashtag goes through an evolutionary process, eventually stabilizing on its least-effort form. The easier recognition of canonical forms is in line with cognitive findings by (Nosofsky and Palmeri 1997) and (Mahowald et al. 2013). An extreme example is the evolution of *#ff*<sup>12</sup> – the most frequent hashtag in our collection, used hundreds of thousands of times. While *#ff* and *#followfriday* are cryptic and incomprehensible to the untrained eye, starting as *#FollowFridays* late 2008, it quickly gained popularity, thus it no longer required much effort to comprehend and turned into *#followfriday*, which in turn evolved to *#FF* and *#ff*.

The evolutionary process of the *#ff* is illustrated in Figure 2<sup>13</sup>. The canonization process is evident by the trend line of *#ff* and *#FF* compared to the other options.

Identifying the evolutionary process and the canonization of some hashtags is not a trivial task and is left for future work. However, this phenomena, to the extent it is present in our data, may be another source of noise affecting our results.

### Institutionalization (Official Hashtags)

Throughout this paper, we assumed that hashtags are introduced by grass roots users and (some of them) organically adopted by the wider community. We hypothesized that the set of constraints (or the weights of the constraints) may change once a hashtag reaches some popularity threshold that makes it easily recognizable in a non optimal form (e.g. *#ff* discussed above). However, not all hashtags are grass

<sup>12</sup>*#ff*, abbreviation for ‘follow Friday’, is used by the Twitter community on Fridays as a means for a tweeter to introduce new tweeters to her followers. A typical *#ff* tweet looks like “*#ff* @user1 @user2 @user3 @user4”.

<sup>13</sup>Our dataset does not contain tweets posted as early as 2008 but the trend can be interpolated easily.

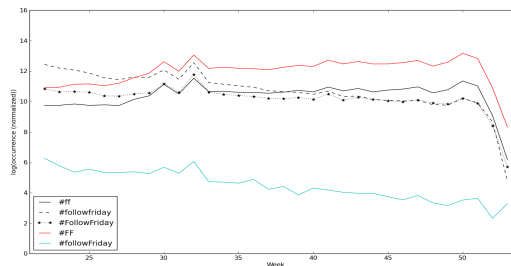


Figure 2: Trend lines of the popular *#followFriday* hashtag in its various incarnations. (Y axis is in logarithmic scale).

Official Hashtag	Hashtag	Counts
HIMYM	HIMYM	139595
	himym	30190
	HowIMetYourMother	1032
gossipgirl	gossipgirl	80483
	GossipGirl	28942
	gossipGirl	220
xfactor	xfactor	325622
	Xfactor	28942
	xFactor	2067

Table 7: Hashtags counts for *How I met Your Mother*, *Gossip Girl* and *X Factor* during 2013 TV season.

roots phenomena. More and more “formal” hashtags are introduced by institutions as twitter gained its huge popularity with the years. Some examples include hashtags of political campaigns such as *#AreYouBetterOff* and *#Forward2012* (Republican and Democrats, respectively in the 2012 presidential campaign) and of TV shows such as *#HIMYM* and *#gossipgirl* (For the shows *How I Met Your Mother* and *Gossip Girl*). These hashtags are aggressively promoted by their creator, either it is a presidential candidate or a TV network. The aggressive promotion in TV commercials, promos and shows as well as in paper ads introduces a strong bias toward using the official hashtags and against the regular constraints.

Checking all hashtags for the 113 most popular TV series aired in the American TV during 2013 we find that in 91.1% of the shows the official hashtags are used extensively comparing to other hashtags that may be more appropriate by our model. Some examples of such hashtags can be found in Table 7. We attribute these numbers to the massive exposure and aggressive promotion campaigns.

Analyzing tens of thousands of hashtags in our corpus, it is hard to verify which hashtags are official. In order to minimize bias induced by aggressively promoted official hashtags we opted to use data that was generated at a time in which Twitter was popular enough to serve as an active arena for the propagation of language while still not discovered by the mainstream media.

We note that campaign architects can use our algorithm in order to devise official hashtags that will be more effective at lower cost.



## Related work

Neologisms are understudied from a computational perspective. Cook and Stevenson show that knowledge of etymology and word formation can be encoded and exploited for automatic identification of neologisms in blog corpora and text messages (Cook and Stevenson 2007; Cook 2010).

The acceptance of a neologism can be viewed as a propagation process. Information diffusion in online social networks has earned much interest in the past decade; however, the main foci of this body of work are the graph topology, the strength of connections between graph nodes and the activity level of individual nodes, see (Kempe, Kleinberg, and Tardos 2003; Chen, Wang, and Wang 2010; Yang and Leskovec 2010; Kleinberg 2010; Romero, Meeder, and Kleinberg 2011), among others. In this work we shift the focus from the graph topology to the linguistic traits that facilitate efficient diffusion of the information.

A few works address some linguistic aspects that affect the information spread. Analyzing Twitter hashtags of three topics, Chuna et al. (2011) find correlation between a hashtag's length and its frequency. Tsur and Rappaport (2012) predict hashtag popularity in a range of time frames given some linguistic features (e.g. length in characters and words, lexicality, semantic class) along with graph topology, and early temporal trends.

While in Tsur and Rappaport each hashtag “competes” for popularity with each and every other hashtag, we believe that small sets of interchangeable hashtags should be addressed separately in order to gain linguistic insights. We therefore propose a ranking task in which we learn the user's preference between similar candidates. Unlike Chuna et al., we analyze thousands of such sets and acknowledge a much broader set of linguistic features. Unlike both Chuna et al. and Tsur and Rappaport, we allow non linear models, hypothesizing that the relations between linguistic constraints are not linear. This hypothesis is well supported by our results.

We also note that our work is loosely inspired by the linguistic Optimality Theory paradigm (Prince and Smolensky 1997) in which observed forms of language are the result of a complex dynamic between competing constraints.

The few quantitative works related to neologism and word evolution are surveyed in the Introduction section. To the best of our knowledge, this is the first computational work to address textual neologisms in the light of competition between constraints.

## Conclusion and Future Work

In this paper we presented a new task – predicting the popularity of hashtags in their immediate context of competing semantically interchangeable hashtags. We view Twitter hashtags as neologisms, and presented a linguistically motivated algorithm for predicting user preferences in using hashtags. The algorithmic framework is a modification of Gradient Boosted Trees adapted for ranking. Our algorithm outperforms a naive, yet strong, baseline and an SVM-rank algorithm. Another major advantage of our algorithmic framework is its interpretability which provides insights to

the underlying processes governing language production.

In future work we will address hashtag ambiguity, and study the socio-linguistic effects of canonization and institutionalization (formal hashtags) on the use of hashtags. Another direction should be the expansion of the semantically interchangeable sets to include more variation. On the methodological side we plan to introduce magnitude preserving metrics to the algorithmic framework.

## References

- Algeo, J. 1977. Blends, a structural and systemic view. *American Speech* 52(1/2):47–64.
- Algeo, J. 1980. Where do all the new words come from? *American Speech* 55(4):264–277.
- Berger, j., and Milkman, L. 2012. What makes online content viral. *Marketing Research* 49(2):192–205.
- Brinton, L. J., and Traugott, E. C. 2005. *Lexicalization and language change*. Cambridge University Press.
- Caruana, R., and Niculescu-Mizil, A. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, 161–168. ACM.
- Chen, W.; Wang, C.; and Wang, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1029–1038. ACM.
- Cheng, J.; Adamic, L.; Dow, P.; Kleinberg, J.; and Leskovec, J. 2014. Can cascades be predicted? In *Proceedings of the 23st international conference companion on World Wide Web*. ACM.
- Chomsky, N. 1976. *Reflections on language*. Temple Smith London.
- Cook, P., and Stevenson, S. 2007. Automagically inferring the source words of lexical blends. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING 2007)*, 289–297.
- Cook, C. P. 2010. *Exploiting linguistic knowledge to infer properties of neologisms*. Ph.D. Dissertation, University of Toronto.
- Cunha, E.; Magno, G.; Comarela, G.; Almeida, V.; Gonçalves, M. A.; and Benevenuto, F. 2011. Analyzing the dynamic evolution of hashtags on twitter: a language-based approach. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, 58–65.
- Danescu-Niculescu-Mizil, C.; West, R.; Jurafsky, D.; Leskovec, J.; and Potts, C. 2013. No country for old members: User lifecycle and linguistic change in online communities. In *Proceedings of WWW*.
- Danescu-Niculescu-Mizil, C.; Gamon, M.; and Dumais, S. 2011. Mark my words! Linguistic style accommodation in social media. In *Proceedings of WWW*, 745–754.
- Deutscher, G. 2006. *The unfolding of language*. Arrow.
- Friedman, J. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38(4):367–378.

- Goldberg, Y., and Orwant, J. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM)*, volume 1, 241–247.
- Grice, H. P. 1975. Logic and conversation. 1975 41–58.
- Hammond, M. 1996. Binariness in english nouns. *Nordlyd* 24:97–110.
- Joachims, T. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 217–226. ACM.
- Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146. ACM.
- Kleinberg, J. 2010. The flow of on-line information in global networks. In *Proceedings of the 2010 international conference on Management of data*, 1–2. ACM.
- Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Memetracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 497–506. Citeseer.
- Lieberman, E.; Michel, J.-B.; Jackson, J.; Tang, T.; and Nowak, M. A. 2007. Quantifying the evolutionary dynamics of language. *Nature* 449(7163):713–716.
- Lindblom, B. 1990. Explaining phonetic variation: A sketch of the h&h theory. In *Speech production and speech modelling*. Springer. 403–439.
- Ma, j.; Sun, A.; and Cong, G. 2013. On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*.
- Mahowald, K.; Fedorenko, E.; Piantadosi, S.; and Gibson, E. 2013. Info/information theory: speakers choose shorter words in predictive contexts. *Cognition* 126(2):313–318.
- Michel, J.-B.; Shen, Y. K.; Aiden, A. P.; Veres, A.; Gray, M. K.; Pickett, J. P.; Hoiberg, D.; Clancy, D.; Norvig, P.; Orwant, J.; et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science* 331(6014):176–182.
- Nosofsky, R. M., and Palmeri, T. J. 1997. An exemplar-based random walk model of speeded classification. *Psychological Review; Psychological Review* 104(2):266.
- Petersen, A. M.; Tenenbaum, J. N.; Havlin, S.; Stanley, H. E.; and Perc, M. 2012. Languages cool as they expand: Allometric scaling and the decreasing need for new words. *Scientific Reports* 2.
- Piantadosi, S.; Tily, H.; and Gibson, E. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences* 108(9):3526–3529.
- Prince, A., and Smolensky, P. 1997. Optimality: From neural networks to universal grammar. *Science* 275(5306):1604–1610.
- Romero, D. M.; Meeder, B.; and Kleinberg, J. 2011. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, 695–704. ACM.
- Schapire, R. E. 1990. The strength of weak learnability. *Machine learning* 5(2):197–227.
- Shannon, C.; Weaver, W.; Blahut, R.; and Hajek, B. 1949. *The mathematical theory of communication*, volume 117. University of Illinois press Urbana.
- Tsur, O., and Rappoport, A. 2012. What’s in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 643–652. ACM.
- Tsur, O.; Littman, A.; and Rappoport, A. 2013. Efficient clustering of short messages into general domains. In *Proceedings of the seventh international AAAI conference on weblogs and social media*.
- Yang, J., and Leskovec, J. 2010. Modeling Information Diffusion in Implicit Networks. In *2010 IEEE International Conference on Data Mining*, 599–608. IEEE.
- Zheng, Z.; Chen, K.; Sun, G.; and Zha, H. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 287–294. ACM.
- Zipf, G. 1949. Human behavior and the principle of least effort.