# A Computational Approach to the Automation of Creative Naming

**Gözde Özbal**
FBK-Irst / Trento, Italy
gozbalde@gmail.com

**Carlo Strapparava**
FBK-Irst / Trento, Italy
strappa@fbk.eu

## Abstract

In this paper, we propose a computational approach to generate neologisms consisting of homophonic puns and metaphors based on the category of the service to be named and the properties to be underlined. We describe all the linguistic resources and natural language processing techniques that we have exploited for this task. Then, we analyze the performance of the system that we have developed. The empirical results show that our approach is generally effective and it constitutes a solid starting point for the automation of the naming process.

## 1 Introduction

A catchy, memorable and creative name is an important key to a successful business since the name provides the first image and defines the identity of the service to be promoted. A good name is able to state the area of competition and communicate the promise given to customers by evoking semantic associations. However, finding such a name is a challenging and time consuming activity, as only few words (in most cases only one or two) can be used to fulfill all these objectives at once. Besides, this task requires a good understanding of the service to be promoted, creativity and high linguistic skills to be able to play with words. Furthermore, since many new products and companies emerge every year, the naming style is continuously changing and creativity standards need to be adapted to rapidly changing requirements.

The creation of a name is both an art and a science (Keller, 2003). Naming has a precise methodology and effective names do not come out of the blue. Although it might not be easy to perceive all the effort behind the naming process just based on the final output, both a training phase and a long process consisting of many iterations are certainly required for coming up with a good name.

From a practical point of view, naming agencies and branding firms, together with automatic name generators, can be considered as two alternative services that facilitate the naming process. However, while the first type is generally expensive and processing can take rather long, the current automatic generators are rather naïve in the sense that they are based on straightforward combinations of random words. Furthermore, they do not take semantic reasoning into account.

To overcome the shortcomings of these two alternative ways (i.e. naming agencies and naïve generators) that can be used for obtaining name suggestions, we propose a system which combines several linguistic resources and natural language processing (NLP) techniques to generate creative names, more specifically neologisms based on homophonic puns and metaphors. In this system, similarly to the previously mentioned generators, users are able to determine the category of the service to be promoted together with the features to be emphasized. Our improvement lies in the fact that instead of random generation, we take semantic, phonetic, lexical and morphological knowledge into consideration to automatize the naming process.

Although various resources provide distinct tips for inventing creative names, no attempt has been made to combine all means of creativity that can be used during the naming process. Furthermore, in addition to the devices stated by copywriters, there



*Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 703–711,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

might be other latent methods that these experts unconsciously use. Therefore, we consider the task of discovering and accumulating all crucial features of creativity to be essential before attempting to automatize the naming process. Accordingly, we create a gold standard of creative names and the corresponding creative devices that we collect from various sources. This resource is the starting point of our research in linguistic creativity for naming.

The rest of the paper is structured as follows. First, we review the state-of-the-art relevant to the naming task. Then, we give brief information about the annotation task that we have conducted. Later on, we describe the model that we have designed for the automatization of the naming process. Afterwards, we summarize the annotation task that we have carried out and analyze the performance of the system with concrete examples by discussing its virtues and limitations. Finally, we draw conclusions and outline ideas for possible future work.

## 2 Related Work

In this section, we will analyze the state of the art concerning the naming task from three different aspects: i) linguistic ii) computational iii) commercial.

### 2.1 Linguistic

Little research has been carried out to investigate the linguistic aspects of the naming mechanism. B. V. Bergh (1987) built a four-fold linguistic topology consisting of phonetic, orthographic, morphological and semantic categories to evaluate the frequency of linguistic devices in brand names. Bao et al. (2008) investigated the effects of relevance, connotation, and pronunciation of brand names on preferences of consumers. Klink (2000) based his research on the area of sound symbolism (i.e. "the direct linkage between sound and meaning" (Leanne Hinton, 2006)) by investigating whether the sound of a brand name conveys an inherent meaning and the findings showed that both vowels and consonants of brand names communicate information related to products when no marketing communications are available. Kohli et al. (2005) analyzed consumer evaluations of meaningful and non-meaningful brand names and the results suggested that non-meaningful brand names are evaluated less favorably than meaningful ones even after repeated exposure. Lastly, cog (2011) focused on the semantics of branding and based on the analysis of several international brand names, it was shown that cognitive operations such as domain reduction/expansion, mitigation, and strengthening might be used unconsciously while creating a new brand name.

### 2.2 Computational

To the best of our knowledge, there is only one computational study in the literature that can be applied to the automatization of name generation. Stock and Strapparava (2006) introduce an acronym ironic reanalyzer and generator called HAHAcronym. This system both makes fun of existing acronyms, and produces funny acronyms that are constrained to be words of the given language by starting from concepts provided by users. HAHAcronym is mainly based on lexical substitution via semantic field opposition, rhyme, rhythm and semantic relations such as antonyms retrieved from WordNet (Stark and Riesenfeld, 1998) for adjectives.

As more naïve solutions, automatic name generators can be used as a source of inspiration in the brainstorming phase to get ideas for good names. As an example, `www.business-name-generators.com` randomly combines abbreviations, syllables and generic short words from different domains to obtain creative combinations. The domain generator on `www.namestation.com` randomly generates name ideas and available domains based on alliterations, compound words and custom word lists. Users can determine the prefix and suffix of the names to be generated. The brand name generator on `www.netsubstance.com` takes keywords as inputs and here users can configure the percentage of the shifting of keyword letters. Lastly, the mechanism of `www.naming.net` is based on name combinations among common words, Greek and Latin prefixes, suffixes and roots, beginning and ending word parts and rhymes. A shortcoming of these kinds of automatic generators is that random generation can output so many bad suggestions and users have to be patient to find the name that they are looking for. In addition, these generations are based on straightforward combinations of words and they do not include a mechanism to also take semantics into account.

### 2.3 Commercial

Many naming agencies and branding firms[1] provide professional service to aid with the naming of new

[1] e.g. `www.eatmywords.com`, `www.designbridge.com`, `www.ahundredmonkeys.com`

products, domains, companies and brands. Such services generally require customers to provide brief information about the business to be named, fill in questionnaires to learn about their markets, competitors, and expectations. In the end, they present a list of name candidates to be chosen from. Although the resulting names can be successful and satisfactory, these services are very expensive and the processing time is rather long.

## 3 Dataset and Annotation

In order to create a gold standard for linguistic creativity in naming, collect the common creativity devices used in the naming process and determine the suitable ones for automation, we conducted an annotation task on a dataset of 1000 brand and company names from various domains (Özbal et al., 2012). These names were compiled from a book dedicated to brand naming strategies (Botton and Cegarra, 1990) and various web resources related to creative naming such as `adslogans.co.uk` and `brandsandtags.com`.

Our list contains names which were invented via various creativity methods. While the creativity in some of these names is independent of the context and the names themselves are sufficient to realize the methods used (e.g. alliteration in *Peak Performance*, modification of one letter in *Vimeo*), for some of them the context information such as the description of the product or the area of the company is also necessary to fully understand the methods used. For instance, *Thanks a Latte* is a coffee bar name where the phonetic similarity between "lot" and "latte" (a coffee type meaning "milk" in Italian) is exploited. The name *Caterpillar*, which is an earth-moving equipment company, is used as a metaphor. Therefore, we need extra information regarding the domain description in addition to the names. Accordingly, while building our dataset, we conducted two separate branches of annotation. The first branch required the annotators to fill in the domain description of the names in question together with their etymologies if required, while the second asked them to determine the devices of creativity used in each name.

In order to obtain the list of creativity devices, we collected a total of 31 attributes used in the naming process from various resources including academic papers, naming agents, branding and advertisement experts. To facilitate the task for the annotators, we subsumed the most similar attributes when required. Adopting the four-fold linguistic topology suggested by Bergh et al. (B. V. Bergh, 1987), we mapped these attributes into phonetic, orthographic, morphological and semantic categories. The phonetic category includes attributes such as rhyme (i.e. repetition of similar sounds in two or more words - e.g. *Etch-a-sketch*) and reduplication (i.e. repeating the root or stem of a word or part of it exactly or with a slight change - e.g. *Teenie Weenie*), while the orthographic category consists of devices such as acronyms (e.g. *BMW*) and palindromes (i.e. words, phrases, numbers that can be read the same way in either direction e.g. *Honda "Civic"*). The third category is the morphology which contains affixation (i.e. forming different words by adding morphemes at the beginning, middle or end of words - e.g. *Nutella*) and blending (i.e. forming a word by blending sounds from two or more distinct words and combining their meanings - e.g. *Wikipedia* by blending "Wiki" and "encyclopedia"). Finally, the semantic category includes attributes such as metaphors (i.e. Expressing an idea through the image of another object - e.g. *Virgin*) and punning (i.e. using a word in different senses or words with sound similarity to achieve specific effect such as humor - e.g. *Thai Me Up* for a Thai restaurant).

## 4 System Description

The resource that we have obtained after the annotation task provides us with a starting point to study and try to replicate the linguistic and cognitive processes behind the creation of a successful name. Accordingly, we have made a systematic attempt to replicate these processes, and implemented a system which combines methods and resources used in various areas of Natural Language Processing (NLP) to create neologisms based on homophonic puns and metaphors. While the variety of creativity devices is actually much bigger, our work can be considered as a starting point to investigate which kinds of technologies can successfully be exploited in which way to support the naming process. The task that we deal with requires: 1) reasoning of relations between entities and concepts; 2) understanding the desired properties of entities determined by users; 3) identifying semantically related terms which are also consistent with the objectives of the advertisement; 4) finding terms which are suitable metaphors for the properties that need to be emphasized; 5) reasoning about phonetic properties of words; 6) combining all this information to create natural sounding neologisms.

In this section, we will describe in detail the work flow of the system that we have designed and implemented to fulfill these requirements.

### 4.1 Specifying the category and properties

Our design allows users to determine the category of the product/brand/company to be advertised (e.g. shampoo, car, chocolate) optionally together with the properties (e.g. softening, comfortable, addictive) that they want to emphasize. In the current implementation, categories are required to be nouns while properties are required to be adjectives. These inputs that are specified by users constitute the main *ingredients* of the naming process. After the determination of these ingredients, several techniques and resources are utilized to enlarge the ingredient list, and thereby to increase the variety of new and creative names.

### 4.2 Adding common sense knowledge

After the word defining the category is determined by the user, we need to automatically retrieve more information about this word. For instance, if the category has been determined as "shampoo", we need to learn that "it is used for washing hair" or "it can be found in the bathroom", so that all this extra information can be included in the naming process. To achieve that, we use ConceptNet (Liu and Singh, 2004), which is a semantic network containing common sense, cultural and scientific knowledge. This resource consists of nodes representing concepts which are in the form of words or short phrases of natural language, and labeled relations between them.

ConceptNet has a closed class of relations expressing connections between concepts. After the analysis of these relations according to the requirements of the task, we have decided to use the ones listed in Table 1 together with their description in the second column. The third column states whether the category word should be the first or second argument of the relation in order for us to consider the new word that we discover with that relation. Since, for instance, the relations *MadeOf(milk, *)* and *MadeOf(*, milk)* can be used for different goals (the former to obtain the ingredients of milk, and the latter to obtain products containing milk), we need to make this differentiation. Via ConceptNet 5, the latest version of ConceptNet, we obtain a list of relations such as *AtLocation(shampoo, bathroom)*, *UsedFor(shampoo, clean)* and *MadeOf(shampoo, perfume)* with the query word "shampoo". We add all the words appearing in relations with the category word to our ingredient list. Among these new words, multiwords are filtered out since most of them are noisy and for our task a high precision is more important than a high recall.

| Relation | Description | # | POS |
|---|---|---|---|
| HasA | What does it possess? | 1 | n |
| PartOf | What is it part of? | 2 | n |
| UsedFor | What do you use it for? | 1 | n,v |
| AtLocation | Where would you find it? | 2 | n |
| MadeOf | What is it made of | 1 | n |
| CreatedBy | How do you bring it into existence? | 1 | n |
| HasSubevent | What do you do to accomplish it? | 2 | v |
| Causes | What does it make happen? | 1 | n,v |
| Desires | What does it want? | 1 | n,v |
| CausesDesire | What does it make you want to do? | 1 | n,v |
| HasProperty | What properties does it have? | 1 | a |
| ReceivesAction | What can you do to it? | 1 | v |

Table 1: ConceptNet relations.

Since sense information is not provided, one of the major problems in utilizing ConceptNet is the difficulty in disambiguating the concepts. In our current design, we only consider the most common senses of words. As another problem, the part-of-speech (POS) information is not available in ConceptNet. To handle this problem, we have determined the required POS tags of the new words that can be obtained from the relations with an additional goal of filtering out the noise. These tags are stated in the fourth column of Table 1.

### 4.3 Adding semantically related words

To further increase the size of the ingredient list, we utilize another resource called WordNet (Miller, 1995), which is a large lexical database for English. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms called synsets. Each synset in WordNet expresses a different concept and they are connected to each other with lexical, semantic and conceptual relations.

We use the *direct hypernym* relation of WordNet to retrieve the superordinates of the category word (e.g. *cleansing agent*, *cleanser* and *cleaner* for the category word *shampoo*). We prefer to use this relation of WordNet instead of the relation "IsA" in

ConceptNet to avoid getting too general words. Although we can obtain only the direct hypernyms in WordNet, no such mechanism exists in ConceptNet. In addition, while WordNet has been built by linguists, ConceptNet is built from the contributions of many thousands of people across the Web and naturally it also contains a lot of noise.

In addition to the direct hypernyms of the category word, we increase the size of the ingredient list by adding synonyms of the category word, the new words coming from the relations and the properties determined by the user.

It should be noted that we do not consider any other statistical or knowledge based techniques for semantic relatedness. Although they would allow us to discover more concepts, it is difficult to understand if and how these concepts pertain to the context. In WordNet we can decide what relations to explore, with the result of a more precise process with possibly less recall.

### 4.4 Retrieving metaphors

A metaphor is a figure of speech in which an implied comparison is made to indicate how two things that are not alike in most ways are similar in one important way. Metaphors are common devices for evocation, which has been found to be a very important technique used in naming according to the analysis of our dataset.

In order to generate metaphors, we start with the set of properties determined by the user and adopt a similar technique to the one proposed by (Veale, 2011). In this work, to metaphorically ascribe a property to a term, stereotypes for which the property is culturally salient are intersected with stereotypes to which the term is pragmatically comparable. The stereotypes for a property are found by querying on the web with the simile pattern "as ⟨*property*⟩ as *". Unlike the proposed approach, we do not apply any intersection with comparable stereotypes since the naming task should favor further terms to the category word in order to exaggerate, to evoke and thereby to be more effective.

The first constituent of our approach uses the pattern "as ⟨*property*⟩ as *" with the addition of "⟨*property*⟩ like *", which is another important block for building similes. Given a property, these patterns are harnessed to make queries through the web api of Google Suggest. This service performs auto-completion of search queries based on popular searches. Although top 10 (or fewer) suggestions are provided for any query term by Google Suggest, we expand these sets by adding each letter of the alphabet at the end of the provided phrase. Thereby, we obtain 10 more suggestions for each of these queries. Among the metaphor candidates that we obtain, we filter out multiwords to avoid noise as much as possible. Afterwards, we conduct a lemmatization process on the rest of the candidates. From the list of lemmas, we only consider the ones which appear in WordNet as a noun. Although the list that we obtain in the end has many potentially valuable metaphors (e.g. *sun*, *diamond*, *star*, *neon* for the property *bright*), it also contains a lot of uncommon and unrelated words (e.g. *downlaod*, *myspace*, *house*). Therefore, we need a filtering mechanism to remove the noise and keep only the best metaphors.

To achieve that, the second constituent of the metaphor retrieval mechanism makes a query in ConceptNet with the given property. Then, all the nouns coming from the relations in the form of *HasProperty(*, property)* are collected to find words having that property. The POS check to obtain only nouns is conducted with a look-up in WordNet as before. It should be noted that this technique would not be enough to retrieve metaphors alone since it can also return noise (e.g. *blouse*, *idea*, *color*, *homeschooler* for the property *bright*).

After we obtain two different lists of metaphor candidates with the two mechanisms mentioned above, we take the intersection of these lists and consider only the words appearing in both lists as metaphors. In this manner, we aim to remove the noise coming from each list and obtain more reliable metaphors. To illustrate, for the same example property *bright*, the metaphors obtained at the end of the process are *sun*, *light* and *day*.

### 4.5 Generating neologisms

After the ingredient list is complete, the phonetic module analyzes all ingredient pairs to generate neologisms with possibly homophonic puns based on phonetic similarity.

To retrieve the pronunciation of the ingredients, we utilize the CMU Pronouncing Dictionary (Lenzo, 2007). This resource is a machine-readable pronunciation dictionary of English which is suitable for uses in speech technology, and it contains over 125,000 words together with their transcriptions. It has mappings from words to their pronunciations

| Input | | Successful output | | Unsuccessful output | |
|---|---|---|---|---|---|
| Category | Properties | Word | Ingredients | Word | Ingredients |
| bar | irish lively wooden traditional warm hospitable friendly | beertender<br>barty<br>giness | bartender, beer<br>party, bar<br>guinness, gin | barkplace<br>barl<br>bark | workplace, bar<br>girl, bar<br>work, bar |
| perfume | attractive strong intoxicating unforgettable feminine mystic sexy audacious provocative | mysticious<br>bussling<br>mysteelious | mysterious, mystic<br>buss, puzzling<br>mysterious, steel | provocadeepe | provocative, deep |
| sunglasses | cool elite though authentic cheap sporty | spectacools<br>electacles<br>polarice | spectacles, cool<br>spectacles, elect<br>polarize, ice | spocleang | sporting, clean |
| restaurant | warm elegant friendly original italian tasty cozy modern | eatalian<br>pastarant<br>peatza | italian, eat<br>restaurant, pasta<br>pizza, eat | dusta<br>hometess | pasta, dust<br>hostess, home |
| shampoo | smooth bright soft volumizing hydrating quality | fragrinse<br>cleansun | fragrance, rinse<br>cleanser, sun | furl<br>sasun | girl, fur<br>satin, sun |

Table 2: A selection of succesful and unsuccessful neologisms generated by the model.

and the current phoneme set contains 39 phonemes based on the ARPAbet symbol set, which has been developed for speech recognition uses. We conducted a mapping from the ARPAbet phonemes to the international phonetic alphabet (IPA) phonemes and we grouped the IPA phonemes based on the phoneme classification documented in IPA. More specifically, we grouped the ones which appear in the same category such as p-b, t-d and s-z for the consonants; i-y and e-ø for the vowels.

After having the pronunciation of each word in the ingredient list, shorter pronunciation strings are compared against the substrings of longer ones. Among the different possible distance metrics that can be applied for calculating the phonetic similarity between two pronunciation strings, we have chosen the Levenshtein distance (Levenshtein, 1966). This distance is a metric for measuring the amount of difference between two sequences, defined as the minimum number of edits required for the transformation of one sequence into the other. The allowable edit operations for this transformation are insertion, deletion, or substitution of a single character. For example, the Levenshtein distance between the strings "kitten" and "sitting" is 3, since the following three edits change one into the other, and there is no way to do it with fewer than three edits: kitten → sitten (substitution of 'k' with 's'), sitten → sittin (substitution of 'e' with 'i'), sittin → sitting (insertion of 'g' at the end). For the distance calculation, we employ relaxation by giving a smaller penalty for the phonemes appearing in the same phoneme groups mentioned previously. We normalize each distance by the length of the pronunciation string considered for the distance calculation and we only allow the combination of word pairs that have a normalized distance score less than 0.5, which was set empirically.

Since there is no one-to-one relationship between letters and phonemes and no information about which phoneme is related to which letter(s) is available, it is not straightforward to combine two words after determining the pairs via Levenshtein distance calculation. To solve this issue, we use the Berkeley word aligner[2] for the alignment of letters and phonemes. The Berkeley Word Aligner is a statistical machine translation tool that automatically aligns words in a sentence-aligned parallel corpus. To adapt this tool according to our needs, we split all the words in our dictionary into letters and their mapped pronunciation to their phonemes, so that the aligner could learn a mapping from phonemes to characters. The resulting alignment provides the information about from which index to which index the replacement of the substring of a word should occur. Accordingly, the substring of the word which has a high phonetic similarity with a specific word is replaced with that word. As an example, if the first ingredient is *bright* and the second ingredient is *light*, the name *blight* can be obtained at the end of

[2]http://code.google.com/p/berkeleyaligner/

this process.

### 4.6 Checking phonetic likelihood

To check the likelihood and well-formedness of the new string after the replacement, we learn a 3-gram language model with absolute smoothing. For learning the language model, we only consider the words in the CMU pronunciation dictionary which also exist in WordNet. This filtering is required in order to eliminate a large number of non-English trigrams which would otherwise cause too high probabilities to be assigned to very unlikely sequences of characters. We remove the words containing at least one trigram which is very unlikely according to the language model. The threshold to determine the unlikely words is set to the probability of the least frequent trigram observed in the training data.

## 5 Evaluation

We evaluated the performance of our system with a manual annotation in which 5 annotators judged a set of neologisms along 4 dimensions: 1) appropriateness, i.e. the number of ingredients (0, 1 or 2) used to generate the neologism which are appropriate for the input; 2) pleasantness, i.e. a binary decision concerning the conformance of the neologism to the sound patterns of English; 3) humor/wittiness, i.e. a binary decision concerning the wittiness of the neologism; 4) success, i.e. an assessment of the fitness of the neologism as a name for the target category/properties (unsuccessful, neutral, successful).

To create the dataset, we first compiled a list of 50 categories by selecting 50 hyponyms of the synset *consumer goods* in WordNet. To determine the properties to be underlined, we asked two annotators to state the properties that they would expect to have in a product or company belonging to each category in our category list. Then, we merged the answers coming from the two annotators to create the final set of properties for each category.

Although our system is actually able to produce a limitless number of results for a given input, we limited the number of outputs for each input to reduce the effort required for the annotation task. Therefore, we implemented a ranking mechanism which used a hybrid scoring method by giving equal weights to the language model and the normalized phonetic similarity. Among the ranked neologisms for each input, we only selected the top 20 to build the dataset. It should be noted that for some input combinations the system produced less than 20 neologisms. Accordingly, our dataset consists of a total number of 50 inputs and 943 neologisms.

| | Dimension | | | |
|---|---|---|---|---|
| | APP | PLE | HUM | SUX |
| 2 | 9.54 | 0 | 0 | 27.04 |
| 3 | 33.3 | 25.34 | 32.77 | 49.52 |
| 4 | 41.68 | 38.6 | 34.57 | 18.77 |
| 5 | 15.48 | 36.06 | 32.66 | 4.67 |
| 3+ | 90.46 | 100 | 100 | 72.96 |

Table 3: Inter-annotator agreement (in terms of majority class, MC) on the four annotation dimensions.

To have a concrete idea about the agreement between annotators, we calculated the majority class for each dimension. With 5 annotators, a majority class greater than or equal to 3 means that the absolute majority of the annotators agreed on the same decision. Table 3 shows the distribution of majority classes along the four dimensions of the annotation. For pleasantness (PLE) and humor (HUM), the absolute majority of the annotators (i.e. 3/5) agreed on the same decision in 100% of the cases, while for appropriateness (APP) the figure is only slightly lower. Concerning success, arguably the most subjective of the four dimensions, in 27% of the cases it is not possible to take a majority decision. Nevertheless, in almost 73% of the cases the absolute majority of the annotators agreed on the annotation of this dimension.

Table 4 shows the micro and macro-average of the percentage of cases in which at least 3 annotators have labeled the ingredients as appropriate (APP), and the neologisms as pleasant (PLE), humorous (HUM) or successful (SUX). The system selects appropriate ingredients in approximately 60% of the cases, and outputs pleasant, English-sounding names in ∼87% of the cases. Almost one name out of four is labeled as successful by the majority of the annotators, which we regard as a very positive result considering the difficulty of the task. Even though we do not explicitly try to inject humor in the neologisms, more than 15% of the generated names turn out to be witty or amusing. The system managed to generate at least one successful name for all 50 input categories and at least one witty name for 42. As expected, we found out that there is a very high correlation (91.56%) between the appropriateness of the

| | Dimension | | | |
|---|---|---|---|---|
| Accuracy | APP | PLE | HUM | SUX |
| micro | 59.60 | 87.49 | 16.33 | 23.86 |
| macro | 60.76 | 87.01 | 15.86 | 24.18 |

Table 4: Accuracy of the generation process along the four dimensions.

ingredients and the success of the name. A successful name is also humorous in 42.67% of the cases, while 62.34% of the humorous names are labeled as successful. This finding confirms our intuition that amusing names have the potential to be very appealing to the customers. In more than 76% of the cases, a humorous name is the product of the combination of appropriate ingredients.

In Table 2, we show a selection of successful and unsuccessful outputs generated for the category and the set of properties listed under the block of columns labeled as *Input* according to the majority of annotators (i.e. 3 or more). As an example of positive outcomes, we can focus on the columns under *Successful output* for the input target word *restaurant*. The model correctly selects the ingredients *eat* (a restaurant is *UsedFor* eating), *pizza* and *pasta* (which are found *AtLocation* restaurant) to generate an appropriate name. The three "palatable" neologisms generated are *eatalian* (from the combination of *eat* and *Italian*), *pastarant* (*pasta* + *restaurant*) and *peatza* (*pizza* + *eat*). These three suggestions are amusing and have a nice ring to them. As a matter of fact, it turns out that the name *Eatalian* is actually used by at least one real Italian restaurant located in Los Angeles, CA[3].

For the same set of stimuli, the model also selects some ingredients which are not really related to the use-case, e.g., *dust* and *hostess* (both of which can be found *AtLocation* restaurant) and *home* (a synonym for *plate*, which can be found *AtLocation* restaurant, in the baseball jargon). With these ingredients, the model produces the suggestion *dusta* which sounds nice but has a negative connotation, and *hometess* which can hardly be associated to the input category.

A rather common class of unsuccessful outputs include words that, by pure chance, happen to be already existing in English. In these cases, no actual neologism is generated. Sometimes, the generated words have rather unpleasant or irrelevant meanings, as in the case of *bark* for *bar*. Luckily enough, these kinds of outputs can easily be eliminated by filtering out all the output words which can already be found in an English dictionary or which are found to have a negative valence with state-of-the-art techniques (e.g. SentiWordNet (Esuli and Sebastiani, 2006)). Another class of negative results includes neologisms generated from ingredients that the model cannot combine in a good English-sounding neologism (e.g. *spocleang* from *sporting* and *clean* for *sunglasses* or *sasun* from *satin* and *sun* for *shampoo*).

## 6 Conclusion

In this paper, we have focused on the task of automatizing the naming process and described a computational approach to generate neologisms with homophonic puns based on phonetic similarity. This study is our first step towards the systematic emulation of the various creative devices involved in the naming process by means of computational methods.

Due to the complexity of the problem, a unified model to handle all the creative devices at the same time seems outside the reach of the current state-of-the-art NLP techniques. Nevertheless, the resource that we collected, together with the initial implementation of this model should provide a good starting point for other researchers in the area. We believe that our contribution will motivate other research teams to invest more effort in trying to tackle the related research problems.

As future work, we plan to improve the quality of the output by considering word sense disambiguation techniques to reduce the effect of inappropriate ingredients. We also want to extend the model to include multiword ingredients and to generate not only words but also short phrases. Then, we would like to focus on other classes of creative devices, such as affixation or rhyming. Lastly, we plan to make the system that we have developed publicly available and collect user feedback for further development and improvement.

## Acknowledgments

The authors were partially supported by a Google Research Award.

[3] http://www.eataliancafe.com/

## References

L. Oliver B. V. Bergh, K. Adler. 1987. Linguistic distinction among top brand names. *Journal of Advertising Research*, pages 39–44.

Yeqing Bao, Alan T Shao, and Drew Rivers. 2008. Creating new brand names: Effects of relevance, connotation, and pronunciation. *Journal of Advertising Research*, 48(1):148.

Marcel Botton and Jean-Jack Cegarra, editors. 1990. *Le nom de marque*. Paris McGraw Hill.

2011. Cognitive tools for successful branding. *Applied Linguistics*, 32:369–388.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. pages 417–422.

Kevin Lane Keller. 2003. *Strategic brand management: building, measuring and managing brand equity*. New Jersey: Prentice Hall.

Richard R. Klink. 2000. Creating brand names with meaning: The use of sound symbolism. *Marketing Letters*, 11(1):5–20.

C Kohli, K Harich, and Lance Leuthesser. 2005. Creating brand identity: a study of evaluation of new brand names. *Journal of Business Research*, 58(11):1506–1515.

John J. Ohala Leanne Hinton, Johanna Nichols. 2006. *Sound Symbolism*. Cambridge University Press.

Kevin Lenzo. 2007. The cmu pronouncing dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

V. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707—710.

H. Liu and P. Singh. 2004. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Gözde Özbal, Carlo Strapparava, and Marco Guerini. 2012. Brand Pitt: A corpus to explore the art of naming. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC-2012)*, Istanbul, Turkey, May.

Michael M. Stark and Richard F. Riesenfeld. 1998. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*. MIT Press.

Oliviero Stock and Carlo Strapparava. 2006. Laughing with HAHAcronym, a computational humor system. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1675–1678. AAAI Press.

Tony Veale. 2011. Creative language retrieval: A robust hybrid of information retrieval and linguistic creativity. In *Proceedings of ACL 2011*, Portland, Oregon, USA, June.