Expanding and Weighting Stereotypical Properties of Human Characters for Linguistic Creativity

Khalid Alnajjar[1] alnajjar@cs.helsinki.fi
Mika Hämäläinen[1] mika.hamalainen@helsinki.fi
Hanyang Chen[2] hanyang.chen@ucdconnect.ie
Hannu Toivonen[1] hannu.toivonen@cs.helsinki.fi

[1]Dept. of Computer Science and HIIT, University of Helsinki, Finland
[2]School of Computer Science and Informatics, University College Dublin, Belfield D4, Ireland

Abstract

Many linguistic creativity applications rely heavily on knowledge of nouns and their properties. However, such knowledge sources are scarce and limited. We present a graph-based approach for expanding and weighting properties of nouns with given initial, non-weighted properties. In this paper, we focus on famous characters, either real or fictional, and categories of people, such as Actor, Hero, Child etc. In our case study, we started with an average of 11 and 25 initial properties for characters and categories, for which the method found 63 and 132 additional properties, respectively. An empirical evaluation shows that the expanded properties and weights are consistent with human judgement. The resulting knowledge base can be utilized in creation of figurative language. For instance, metaphors based on famous characters can be used in various applications including story generation, creative writing, advertising and comic generation.

Introduction

Creation and interpretation of figurative language are difficult tasks to tackle by computational means. This is due to the fact that the meaning of a figurative utterance cannot be deduced by the compositional semantic meaning of the words and syntax used, but the meaning rather lies in the pragmatics. This complicates the use of figurative language in computational creativity applications, because pragmatic meaning is open to human interpretation. Therefore when exposed to a computationally generated utterance, the reader might attribute more to it than what is actually there. That is why, when approaching figurative language from the point of view of computational creativity, it is important that the creative system knows what kind of a message is likely to be conveyed pragmatically by a certain figurative sentence.

The relations between nouns, both characters and categories, and their linked adjectival properties can be used in various creative tasks such as generating and interpreting metaphors, which are a common figurative language device. In a simplified form, by knowing that a given property is strongly related to a noun A, we can construct a nominal metaphor that indirectly conveys the meaning of another noun B having this property by stating "B is A". However, such a metaphor doesn't convey the meaning on the semantic level, but rather pragmatically. Therefore, by knowing the stereotypical adjectival properties of the nouns used in the metaphor, we gain access to its pragmatic interpretation.

Consider, as an example, the sentence "Britney Spears is a cat". Following the terminology of Richards (1936), in this metaphor "cat" is known as the *vehicle*, the noun whose property is reflected to the *tenor*, "Britney Spears".

In order to generate such a metaphor, it is important to know the stereotypical properties the characters have. For example, to understand the previous example as equivalent to "Britney Spears is wild", one must know that wildness is a strong property of *cats*. It is important to bear in mind that, because we are dealing with figurative language, the interpretation given as an example is not the only possible one. Therefore, it is also important to know how strongly properties are linked to nouns in order to construct metaphors that are likely to carry the intended meaning, or to reach the most plausible interpretations.

In this paper, we propose a computational method for obtaining a list of stereotypical properties for characters by expanding their given properties, using the NOC list (Veale 2016) as our starting point. This is done for famous characters (e.g. *Britney Spears*) as well as the categories these characters belong to (e.g. Singer). Our method expands a given initial set of properties provided by the NOC list with additional knowledge gathered from the internet. Our methods also weight the noun–property associations, i.e., they estimate how strongly a property is associated with a given noun.

In this paper, only the properties that are strongly linked to nouns are considered stereotypical, meaning that our approach will eradicate the properties that are in the semantic penumbra of the noun they are linked to. The word stereotype is used here in similar fashion both in the case of categories and characters, referring to the most descriptive properties, as opposed to the use of the word stereotype exclusively when describing categories of people with a prejudiced connotation.

This work is motivated by computational creativity, with the aim of providing tools to create and interpret figurative language. The method described in this paper, or its results which are publicly available, can be used as an auxiliary tool in systems such as in natural language generators to substitute literal expressions with figurative ones while still retaining the original semantic meaning. In other words, this work provides a piece to the larger puzzle of computational creativity in the context of figurative language, both in its interpretation and generation.

This paper is structured as follows. After briefly reviewing related work, we give an overview of how the initial data is obtained for characters and categories, together with their properties and the limitations the initial data has. We then describe methods for (1) expanding the set of properties for characters and categories, (2) computing weights for the noun–property pairs, and (3) filtering out noun–property pairs with low weights. We continue by reporting on a crowd-sourced evaluation of this expansion, and then discuss the results.

Related Work

The motivation of this paper is metaphors where the tenor and vehicle are either characters or categories of people. Characters, in this case, are famous people from the real or fictional world such as *Albert Einstein* and *Batman*. While characters are all proper nouns, categories are common nouns (such as hero, scientist) that can be used to categorize people.

The work presented in this paper builds upon the foundations of a system called Thesaurus Rex (Veale 2013). In the same way as Thesaurus Rex links nouns to properties, our approach will link both characters and categories to adjectival properties.

As Searle (1958) points out, the whole semantics of a proper noun poses problems far beyond those of common nouns. From the point of view of type-token distinction (Peirce 1974), common nouns can refer to an entire type (e.g. in "dog is man's best friend" the word dog refers to dogs in general), whereas proper nouns refer to tokens (e.g. there's only one *Albert Einstein*).

Constructing meaningful metaphors and interpreting them requires knowledge about the tenor and vehicle, and how they interact with each other. This is done by looking into their shared properties. In the context of this paper, a property can be understood as an adjective that is stereotypically associated with a noun.

Metaphor Magnet (Veale and Li 2012) is based on the notion that stereotype expansion and property overlap are the key components in interpreting metaphors. Nouns that are used both as tenor and vehicle are first expanded with a list of stereotypical properties usually related to the nouns in question. These stereotypical properties are extracted from Google n-gram data by linguistic patterns such as "NOUN_1 is [a] NOUN_2". After this step, the union of the properties related to a noun and its associated stereotypical properties are attributed to the noun. Properties that are in adjectival form, VERB+*ing*s and VERB+*ed*s, are gathered from the internet by applying a different set of linguistic patterns. However, this data was not used in its raw form to build the two knowledge bases, but rather filtered manually. The properties a metaphor conveys are understood as the intersection of the properties of the tenor and those of the vehicle. However, *Metaphor Magnet* includes a limited coverage of characters and their properties (e.g. *Albert Einstein* has only 3 properties (*educated*, *trustworthy*, and *probing*), which makes generating and interpreting metaphors containing *Albert Einstein* infeasible.

Our motivation for this work is that metaphor generation is a knowledge hungry task. Existing metaphor processing methods for metaphor identification, interpretation, and generation rely on a huge amount of knowledge. In the field of distributional semantics, a lot of research has been done to extract word relations from large corpora, e.g., to group words automatically into semantically related categories, using methods such as Word2Vec (Mikolov et al. 2013) or LSA (Landauer and Dumais 1997). Such semantically grouped words can include nouns and adjectives if they co-occurred together in a corpus within the same context. However, these methods generalise far too much for our needs. In expanding properties, we are only interested in the adjectives that are descriptive for a given noun, i.e. not all the adjectives co-occurring with it. In addition, proper nouns rarely co-occur in text with stereotypical adjectives describing them. Therefore, such general distributional semantics approaches cannot be used in the context of this paper.

Obtaining Initial Knowledge

We employ two resources for obtaining initial knowledge of characters and categories along with their properties: Veale's (2016) *NOC List (Non-Official Characterization list)* is used to obtain initial sets of properties related to characters; regarding properties of categories, we utilize the output of an information extraction technique described by Veale and Hao (2008a).

In the rest of this section, we explain how the NOC list is used in our approach and how the initial stereotypical properties of categories are obtained.

The Non-Official Characterization list *The NOC List (Non-Official Characterization list)* (Veale 2016) is a rich resource containing myriad information about 804 famous characters, both real and fictional. The NOC list contains so called positive and negative talking points for each character. These simply mean positive and negative adjectives that describe the character in question. In this paper, we only use the talking points and categories from the NOC list, leaving all the other information in the NOC list aside due to its irrelevance to the problem we are tackling.

Talking points of characters are used as their stereotypical properties. Positive talking points in the NOC list include words such as *funny, convincing, wise* and *powerful*, while examples of negative talking points are *bossy, inhuman, evil*, and *fat*. The talking points are not necessarily true about the characters, instead they are properties that people commonly would associate with these characters when thinking of them. In total, there are 1983 unique properties in the NOC list.

Stereotypical Properties of Categories The NOC list also contains categories of characters. They are often occupations (e.g. Actress and Scientist), but can also refer to other kinds of social groups (e.g. Child and Bully). In the

NOC list, categories do not have associated properties, because they are only used to provide the character entries with more information, i.e. they are not independent entries in the same way character entries are. On average, a character has three categories. Overall, there are 449 unique categories in the NOC list.

The initial stereotypical properties for all the categories mentioned in the NOC list were obtained by the approach described by Veale and Hao (2008b; 2008a; for the datasets, see end of this paper). The Google Search API was used to mine the web for similes with the pattern "as *ADJ* as a|an *NOUN*" with the hypothesis that an adjective *ADJ* is potentially a stereotypical property of a category *NOUN*.

Human judges were asked to annotate whether the properties were meaningful in an empty context to ensure that the properties were of a high quality and to filter out any noisy properties. Empty context here refers to the notion that the properties should make sense even when no additional cues about the context than the property and category themselves are provided.

Out of the 449 categories in the NOC list, the described approach only retrieved properties for 336 categories. This is due to various reasons that we will return to in Discussion.

Resulting Knowledge In the obtained initial knowledge base, on average, a character has 11 stereotypical properties, whereas a category has 25 stereotypical properties. The initial knowledge base has two shortcomings that the rest of this paper will tackle.

First, characters naturally have more than 11 properties. Comparing characters solely based on their properties provided in the NOC list limits the operation as some of the properties that are descriptive of them might not be stated directly in the knowledge base (e.g. *Batman* is *adventurous*).

Second, these stereotypical properties are not weighted. In other words, there is no way of telling whether a character or category is more strongly related to a given property than to another one (e.g. is *Stewie Griffin* more *evil* or more *intelligent*?).

Expanding and Weighting Properties

We expand the sets of stereotypical properties of characters and categories and weight both the initial and the newly inferred properties.

In the expansion phase, we infer new properties of a noun based on the initial knowledge base. For instance, people having the property *brave* are typically considered *adventurous*, and so are *bold, strong, agile* and *resourceful* people. As a result, *Batman* should also be seen *adventurous* as his initial knowledge states that *brave, bold, strong, agile* and *resourceful* are some of his stereotypical properties.

In terms of weighting these properties, our hypothesis is that the more knowledge there is to back up a given claim (such as that *Batman* is *brave*), the higher the weight should be.

Both the expansion and weighting of properties are based on viewing properties in a network of their mutual associations. We start by explaining the methodology of establishing the network, then provide the algorithm that assigns and weights new properties to existing characters based on their initial knowledge. Thereafter, we describe how weak properties are pruned.

Construction of a Property Network In order to predict more stereotypical adjectival properties for nouns, both characters and categories, we construct an undirected weighted network of properties from large corpora. The network is initialized with seed properties from the initial knowledge base.

We use Veale's (2011) neighbouring properties dataset (see end of this paper) to obtain links between properties. Veale used the simile pattern "as *p* and * as" as in "as sweet and * as" to retrieve neighbouring properties of *sweet*. The used search engine, Google, returned matching results, e.g. "as sweet and creamy as" and "as sweet and moist as", along with the frequencies of these phrases. Veale additionally normalized the frequencies. In total, 8644 properties are interconnected with other properties in the dataset.

Using the above-described Veale's (2011) dataset, we construct an undirected weighted graph $G = (V, E, w)$. The set of all retrieved properties constitutes the set V of nodes. The set $E \subset V \times V$ of edges is obtained as all pairs of properties in the dataset; we consider edges undirected/symmetric. The weight $w(p_i, p_j)$ of an edge $(p_i, p_j) \in E$ is obtained from the dataset.

We use $N(p)$ to denote the set of properties adjacent to p, i.e., $N(p) = \{p_j \mid (p, p_j) \in E\}$. We use notation $P(n)$ to refer to the stereotypical properties of a given noun n in the initial knowledge base. The set of all properties known in the initial knowledge base is denoted by $P = \cup_n P(n)$.

Finding and Weighting Related Properties The main objective here is to expand the initial set of properties associated with nouns and weight them. To expand the properties of a given noun n, we iterate over all the properties in our knowledge base, P, and examine their relevance to the input noun n based on the property network constructed above.

Given a noun n, we consider one property $p \in P$ at a time. We find out which other properties are related to both n and p, supporting their mutual relationship:

$$R = P(n) \cap N(p).$$

These supporting properties are used to compute a weight $W(n, p)$ for the stereotypicality of property p for noun n:

$$W(n, p) = \sum_{r \in R} w(r, p).$$

We use uppercase $W(\cdot)$ for the resulting weights of stereotypical properties to distinguish them from the weights between properties. Recall that property pairs have weights (in the network constructed above) but noun-property pairs do not (in the initial knowledge base).

In the special case that $p \in P(n)$, i.e., our initial knowledge already indicates that p is a stereotypical property of noun n, we give p some extra weight:

$$W(n,p) = C + \sum_{r \in R} w(r,p), \quad (1)$$

where C is a positive constant. We define it as $C = mc$, where $m = \max(w(\cdot))$ is the maximum weight of an edge, and $c = 3$ is a constant which depends on how one wants to amplify these special cases. We empirically chose $c = 3$ as a lower value would not have any noticeable effect on the weightings, and a higher value resulted in having the properties in the knowledge base be the highest.

Pruning Weak Properties For any noun n, the method described above yields a weighted list of new properties. Some of the properties may only be weakly related to the noun, however. We therefore only keep the best of them.

First, we only keep the top 20% of the properties for each noun (a character or category).

Second, the final weight $W(n,p)$ has to be greater than m, the maximum of any single weight in the property graph. This is to ensure that every new property is supported by at least two related properties, or more if their weights are smaller.

Third, once all weights are calculated for all nouns, we filter out any property that is not linked to at least two nouns. The rationale is that such properties are not helpful when comparing nouns to produce metaphors.

Results of Expansion and Weighting The described approach found and weighted new properties for 99% of characters (793 out of 804) and 82% of categories (276 out of 336). As a result of all these steps, 63 and 132 new properties were added on average to each character and category, respectively. In addition, all existing properties of nouns were given weights. The expanded and weighted noun-property pairs are publicly available (see end of this paper).

Examples We illustrate the results of property expansion with two real examples.

First, consider the person *Britney Spears*. Her initial properties in the NOC list are *tacky, wacky, sleazy, pretty, burnt-out, energetic* and *sultry*. Our expansion method inferred 29 additional properties for her. Added properties with the highest weights are *sexy, weird, young* and *silly*, while added properties with the lowest weights are *wild, vigorous, shiny, entertaining, skilled, enthusiastic* and *imaginative*. The method aims to only find stereotypical properties, so it considers all these properties for *Britney Spears*, including the ones with lower weights but still above the defined threshold. I.e., *wild* is related but less central to the stereotypical view of *Britney Spears* than what the strong properties are.

As a second example, consider one of *Britney Spears*' categories, namely Singer. Initially it had 21 properties, and the expansion added another 86 properties. Among all the properties, the highest weights are assigned to *expressive, melodic, artistic, lyrical, musical, entertaining, tuneful* and *gifted*. The lowest weights yet above the threshold, on the other hand, are for *energetic, concise, capable, fine, harmonious, soulful* and *humorous*.

Evaluation

We next evaluate the quality of the expanded and weighted set of noun-property pairs. The evaluation is carried out using human judges, crowdsourced through the CrowdFlower[1] platform. In this section, we explain our evaluation setup and the data collected followed by the results.

Evaluation Setup

The goal of this evaluation is to validate the quality of the expanded and weighted set of properties. We do this by empirically checking if the properties and their weights correspond to what people think of them in conjunction of a given noun.

To limit the expense of the evaluation, we used a random sample of noun-property pairs. First, we randomly selected 25% of all nouns (188 characters and 73 categories, in total 261 nouns) to be evaluated. Then we selected four test properties for each noun, seeking for a diverse set of properties to evaluate. We selected one strong property (with a high weight), one "weak" property (with a low weight but still considered substantially related by the method) and two random properties that are not in the expanded set.

More specifically, we divided the noun's expanded set of properties into four equal-width bins based on their weights. We then selected one property at random from the highest bin and one from the lowest bin. For some nouns this process fails to work as intended. If the property expansion was unsuccessful, then the noun only has the initial properties, and they can all have equal weights. In these cases, both selected properties are considered strong and there is no weak property.

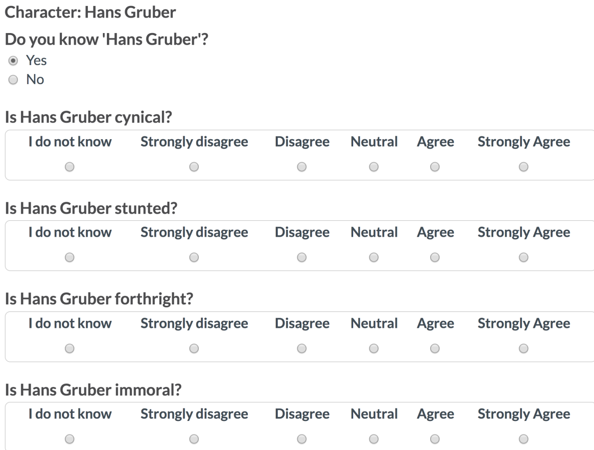
For every noun to be evaluated, we asked judges to rate the four selected properties on a 5-level Likert scale (Figure 1). We used five judges per noun as a trade-off between cost of evaluation and amount of data obtained. We assigned value 1 to "strongly disagree", 2 to "disagree", etc., and eventually 5 to "strongly agree". For each noun-property pair, we then used the average score from the (up to) 5 judges as the score of that pair. We compared the weights given to noun-property pairs by the proposed method to the scores given by the human judges.

The null hypothesis in our tests is that all four properties for each noun come from the same distribution, i.e., the expanded properties effectively are random and the weights do not relate to the strength of noun-property association. The values of weights and scores from judges are not directly comparable due to their different ranges; the statistics we will be using do not assume they are comparable.

Evaluators were allowed to skip nouns they did not know, e.g. a character in a movie they had not seen, by choosing "No" for the first question. In fact, the properties of a noun are shown only if the evaluator knew the noun. If they knew the noun, they were still allowed to indicate that they did not know whether the noun had a given property, as in Figure 1.

[1] https://make.crowdflower.com

Figure 1: An example of a crowdsourced questionnaire


Character: Hans Gruber

Do you know 'Hans Gruber'?

- Yes
- No

Is Hans Gruber cynical?

I do not know	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

Is Hans Gruber stunted?

I do not know	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

Is Hans Gruber forthright?

I do not know	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

Is Hans Gruber immoral?

I do not know	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

The judges were limited to English-speaking countries (Australia, Canada, Ireland, New Zealand, United Kingdom and United States), and were required to have English as a language they speak in their profile. This limitation is enforced because characters in the *NOC list* are largely from Western culture and the language of all the knowledge is English. Moreover, some properties in the initial knowledge bases and the constructed property network are not commonly known even by English speakers, such as matricidal, duplicitous and loquacious.[2]

We did not attempt to remove likely crowdsourcing scammers as this would be difficult due to the subjectivity of how strongly nouns and properties are related. The effect of this decision is that the data is likely to contain additional noise from random entries by scammers or negligent judges.

Data

We obtained evaluations for 261 nouns (and their 4 properties) from 5 judges each, i.e., we had in total 1305 judgements of nouns.

Almost one third (31%) of these judgements were skipped evaluations where the judge indicated they did not know the character or category and thus did not score its properties. This is a relatively high rate and might be inflated due to scammers; however, the number also suggests that it was useful to allow skipping unknown nouns in order to reduce random answers and noise in the actual scores. The average number of evaluators a given noun had is 3.5.

In the following analysis, we ignore nouns that had just one or two evaluators and only consider the 199 nouns (out of 261) which had at least three evaluators. Among these 199 nouns, 127 (64%) are characters and 72 (36%) are categories. These characters are further divided to 93 real and 34 fictional characters.

The judges could also answer that they did not know if the noun had a given property. Overall, 32% of noun-property pairs received the "I do not know" answer (in all 199 considered nouns). Ignoring those two noun-property pairs that only received "do not knows", the total number of evaluated noun-property pairs for the 199 nouns is 794.

For each of the remaining 794 noun-property pairs, we have one to five Likert scores from the human judges. As mentioned above, we map the answers to values from one to five, and take the average of the answers as the score of the noun-property pair.

The inter-judge agreement on the 794 noun-property pairs by the 32 judges, using Krippendorff's alpha measure, is 0.47.

Results

We now consider measures of how well the proposed method performed in its tasks. We first see if it can successfully identify related properties. We then consider three subtly different measures of stereotypicality of noun-property pairs and their correlation with the weights assigned by the proposed method: (1) the mean score as a direct measure of stereotypicality, (2) the standard deviation of the score as a measure of judge agreement (a proxy for stereotypicality), and (3) the number of cases when judges did not know if the noun had the property (a proxy for the inverse of stereotypicality).

Identification of Related Properties The first sub-goal of the method was to find new properties related to given nouns (without weighting them yet). We evaluated how well the method performed in this task using a two-sample permutation test for equal means. We took all new noun-property pairs that were related according to the method as one set, and contrasted them to all random noun-property pairs. The alternative hypothesis was that the mean of scores among the related properties would be higher than among the random properties. The null hypothesis of equal distributions was materialized using 10^7 random permutations of the mean scores across the two sets.

The observed difference between means was higher in the data than in any of the random permutations, yielding $p \approx 10^{-7}$. This statistically highly significant difference between the two sets indicates that the method can successfully identify new related properties using the initial set of properties and an automatically acquired network of related properties. It should be noted, however, that this is more a measure of precision than recall, i.e., the newly found properties tend to be stereotypical for the noun, but there is little information of how many truly stereotypical properties go unnoticed by the method.

Noun-Property Score Let us next take a look at the scores and weights of noun-property pairs. Noun-property pairs with higher scores are likely to be more stereotypical. For simplicity, we pool all nouns together and consider together their strong properties as one set, weak properties as one set, and random properties as one set. In this and all later experiments, where we consider weights of noun-property pairs, we include both the initial pairs and the expanded ones.

The mean scores are given in Table 1. The strong properties have a mean score of 4.13, weak properties have 3.60

[2]Based on the word's difficulty index on Dictionary.com

Table 1: Mean and sample standard deviation of evaluation scores, and number of noun-property pairs evaluated

	Strong Property			Weak Property			Random Properties		
	μ	SD	n	μ	SD	n	μ	SD	n
Categories	4.28	0.64	77	3.43	0.87	66	2.80	0.96	144
Real Char.	3.95	0.92	94	3.62	0.85	92	2.79	0.96	185
Fictional Char.	4.27	0.62	35	3.89	0.75	33	2.88	0.89	68
Total	4.13	0.79	206	3.60	0.85	191	2.81	0.94	397

Table 2: Percentage of noun-property pairs that were rated as "do not know" the among evaluated noun-property pairs

	Strong Property	Weak Property	Random Properties
Categories	20%	30%	76%
Real Characters	19%	21%	54%
Fictional Characters	6%	24%	54%
Total	17%	25%	62%

and random properties 2.18. This indicates a clear general agreement between the human judges and the weights given by the system: the strong properties have on average 0.53 units higher scores than weak properties, and even 1.32 units higher than random properties. The scores of random properties show that judges either disagreed that a given noun has the property or found the association neutral.

For more informative statistical insight on the relation between noun-property weights and evaluation scores, we measured their correlation by simply pooling all noun-property pairs together. The random properties have no weight assigned by the system; for the test here we assumed they have zero weight. This is a very crude approach but helps us gain some insight into the correlation. The Pearson correlation coefficient is $r = 0.48$, with $p \approx 10^{-45}$. The correlation coefficient is not strong (possibly partially due to the simple approach), but the p-value indicates that the correlation is statistically highly significant and not just a random effect.

We also measured the correlation between scores and weights among the related properties only, i.e., ignoring the random properties. Pearson correlation coefficient there is $r = 0.30$ ($p \approx 10^{-9}$) suggesting that it is easier to separate random properties from related properties than strong properties from weak ones. However, the correlation between scores and weights is statistically highly significant also just among the related properties.

Standard Deviation of Scores Additional standard deviations of the scores (Table 1) can provide insight to the degree of agreement between judges. We can see that strong properties are typically more agreed on (have smaller standard deviation); however, in the case of real characters judges seem to have had slightly diverse opinions. Weak properties have higher standard deviation than strong properties, and random properties even larger, indicating less agreement and lower stereotypicality for them.

Unknown Properties In addition to the scores from human judges, we have a complementary measure of stereotypicality: how many judges knew if the noun had the property? Consider a property that has a high numerical score but was not known to be a property of the noun by many judges – such a property can not be considered very stereotypical for the noun.

Table 2 shows the percentage of noun-property pairs that were *not* evaluated by judges because they did not know whether such noun has a given property. We notice a marked increase of this number for random properties, as can be expected. Asked about a property that a character is not specifically known for, a valid answer is to say that one does not know if the character has that property. The number of unknown properties was also higher for the weak properties than for strong properties, indicating higher stereotypicality for the strong properties.

An interesting observation is that fictional characters are very well known for their strong properties (only 6% of "do not knows" vs. 19% for real characters and categories). This is probably due to the fact that fictional characters tend to have more distinctive and emphasized properties than real people; they thus seem to lend themselves better for figurative language such as metaphors.

Discussion

We have proposed and evaluated a method for expanding and weighing sets of properties of characters or categories. The empirical results, based on crowdsourcing, indicate that the method is able to identify new related properties, and to weight initial and new properties to reflect how stereotypical they are for the given noun. A number of issues were encountered during the process, however. We next discuss these issues, as well as possible applications and extensions of the proposed method.

Analysis of Problems in the Method There are three types of problems this method faced: (1) finding results matching a linguistic pattern, (2) lack of sufficient evidence to expand the knowledge base, and (3) limited initial knowledge base.

The *NOC list* contains 449 unique categories; however, for 113 categories, retrieving their properties using "as *ADJ* as a|an *NOUN*" was not successful. This is a problem of type 1 and is due to two main reasons. The first is that the category is a compound word describing another category, such as *Petty Criminal, Roman Gladiator* and *AI Program Villain*. The other reason is that some categories are not commonly used on the internet in the queried pattern, e.g. Symbologist, Lexicographer and Hyperchondriac.

The approach for expanding properties was unable to expand the properties of 11 characters and 60 categories. Regarding characters, the expansion typically failed because there were few links to the character's initial properties. An example of such a case is *Tiger Woods* – a golf player – who has five properties that are not available in the constructed property network, namely *philandering, field-topping, world-beating, highly-paid, long-driving*. This is also a type 1 problem as the links in the network were obtained from the simile pattern "as *p* and * as". Additionally, his remaining four properties, *promiscuous, unfaithful, athletic, and masterful*, are not strongly enough related to each other to infer new links with a weight higher than our defined threshold m, a problem of type 2.

The above two factors also affect the expansion of categories' properties. In addition, some categories have a very limited number of properties retrieved for them in the initial knowledge base, i.e. a problem of type 3. For instance, the categories Linguist and Frontiersman have only one stereotypical property linked to them which is *fluent* and *adventurous*, respectively.

Hence, this approach is expected to work when nouns have a sufficient number of properties (at least ~ 5) that are related to each other and exist in the constructed network. In our case study, this seems to have been the case for most characters and categories. In case where this is not feasible, the pruning conditions can be made lenient to result in higher coverage (e.g. selecting top 50% instead of 20% or reducing the threshold to $m/2$). This can add noise, however, and it requires more experimentation to find out what the exact effects would be.

Applications A direct use case for the proposed approach is in situations where a wider range of properties are required to perform a creative task but only relatively small number of properties are available at hand. For instance, Meta4meaning (Xiao et al. 2016) – a creative corpus-based system for interpreting metaphors – has shown good results of how a creative system can produce interpretations similar to humans. Nevertheless, the system was unable to interpret some metaphors (e.g. "the woman is a cat") due to reasons including that a given noun (*woman*) was not associated with a desired property (*wild*).

Our approach for expanding the list of properties of a noun can be employed in such a scenario. The noun *woman* is not on the NOC list, so we use as examples two specific women instead. Consider the expression "Britney Spears is a cat" and its possible metaphorical meanings. One approach to find potential such meanings is to look at the shared properties of *Britney Spears* and of *cat* (and then pick some of those based on various criteria (Xiao et al. 2016)).

The intersection between the expanded properties of *Britney Spears* and *cat* include properties such as *energetic, spry, vigorous, wild, etc.*, all possible interpretations of the metaphoric expression.

The shared properties between *Hillary Clinton* and *cat*, in turn, include words such as *smart, independent, intelligent, etc*, which are possible metaphorical meanings of the expression "Hillary Clinton is a cat".

Veale (2016) outlines how the NOC list can be used in metaphor generation in the case of characters. In his paper, metaphors are represented as concept pairs that are constructed by using multiple overlapping properties, such as "Hillary Clinton could be Princess Leia: driven yet bossy". His approach provides multiple recommendations for possible metaphors. For a tenor, such as *Tony Stark* (Ironman) and a desired property, such as *rich*, the system outputs possible characters to be used in a metaphor, for example "Tony Stark is Bruce Wayne". The extended weights from our approach can be used to generate these kinds of metaphors in a richer way, since our system provides more knowledge about the stereotypical properties and their weights. This could directly be tested out, for example, with the metaphor generation algorithm proposed by Veale and Li (2012).

The proposed approach is valuable also in creative systems requiring an input from the user, whether they are co-creative systems or not. An example of such a case is generating creative slogans (using BrainSup (Özbal, Pighin, and Strapparava 2013) for instance). Users of these systems are expected to specify the target words such as the brand name and its essential properties to highlight. Our property expansion approach can be utilized in this context to expand the initial set of properties input by the user and weight them to improve the slogans generated.

Properties of Characters in the Context of a Category There are various possible ways to extend the proposed method; we here discuss one interesting avenue that could easily be implemented on top of the current method.

Sometimes an important aspect in metaphors is to know how strong a given property is to a noun when examined from the point of view of a given domain or category. For instance, the weights of the stereotypical property *arrogant* of *Tony Stark* when seen as Hero should be different than when seen as Billionaire.

We hypothesize that such cases can probably be handled by a simple generalization to the definitions of this paper. Assume that n is a character, c a category and p a property. The supporting set of properties is then simply constrained to those that are also properties of category c:

$$R = P(n) \cap N(p) \cap P(c).$$

Equation 1 can then be applied as before. Validation of this technique is a topic for future work.

Conclusions and Future Work

In this paper, we have presented a way of expanding a given initial set of adjectival properties for nouns to cover a wider range of their stereotypical properties, and of weighing the properties. We have successfully applied this approach both in the case of common nouns (categories) and proper nouns (characters). We also conducted an evaluation with human judges to verify the quality of the results obtained by our proposed method.

Based on the new knowledge we constructed about characters and their linked properties, future research can be conducted on computational linguistic creativity, such as metaphor interpretation and generation. An evaluation of metaphors generated using the properties and weights produced by the proposed method would also give additional insight into the quality and usefulness of the results of this paper. Such an evaluation could also inform us about whether metaphors including proper nouns are seen by people in a similar fashion as metaphors only consisting of common nouns. Based on our results, fictional characters look especially promising for metaphors since their stereotypical properties tend to be well known (cf. Table 2).

We have only discussed the expansion of a list of adjectival properties for nouns in this paper. However, the expansion of the nominal components, i.e. categories and characters has been left aside. Given that the origin of the nouns, namely the NOC list, is hand crafted and currently the only way of expanding it is by a laborious manual process, it would be interesting to see in the future if our approach can be used to expand the nominal knowledge as well, for instance, by altering the linguistic patterns.

A possible future direction for this research is to expand it to multiple languages. From a theoretical point of view, there is nothing heavily language dependent that would hinder the adaptation of this method to different languages. Since our approach deals with stereotypes which are known to be socially constructed and thus culturally dependent, this method, in the context of multiple languages, could shed more light on stereotypical beliefs in different cultures.

Datasets

The datasets used as input and produced as output by the methods described in this paper are publicly available at `https://github.com/prosecconetwork/ThesaurusRex/`.

Acknowledgements

This work has been supported by the Academy of Finland under grant 276897 (CLiC) and by the European Commission coordination action PROSECCO (Promoting the Scientific Exploration of Computational Creativity; PROSECCO-network.EU). The authors would like to thank Tony Veale for providing the data sets and inspiration for this work.

References

Landauer, T. K., and Dumais, S. T. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review* 104(2):211–240.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. BrainSup: Brainstorming Support for Creative Sentence Generation. 1446–1455. Sofia, Bulgaria: Association for Computational Linguistics.

Peirce, C. S. 1974. *Collected papers of Charles Sanders Peirce*, volume 2. Harvard University Press.

Richards, I. A. 1936. *The Philosophy of Rhetoric*. London: Oxford University Press.

Searle, J. R. 1958. Ii.proper names. *Mind* 67(266):166–173.

Veale, T., and Hao, Y. 2008a. Enriching wordnet with folk knowledge and stereotypes. In *In Proceedings of the 4th Global WordNet Conference*, 453–461. Szeged, Hungary: Juhász Press Ltd.

Veale, T., and Hao, Y. 2008b. Talking points in metaphor: A concise usage-based representation for figurative processing. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, 308–312. Amsterdam, The Netherlands: IOS Press.

Veale, T., and Li, G. 2012. Specifying viewpoint and information need with affective metaphors: A system demonstration of the metaphor magnet web app/service. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, 7–12. Jeju Island, Korea: Association for Computational Linguistics.

Veale, T. 2011. Creative language retrieval: A robust hybrid of information retrieval and linguistic creativity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 278–287. Stroudsburg, PA, USA: Association for Computational Linguistics.

Veale, T. 2013. A service-oriented architecture for computational creativity. *Journal of Computing Science and Engineering* 7(3):159–167.

Veale, T. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, 34–41. San Diego, California: Association for Computational Linguistics.

Xiao, P.; Alnajjar, K.; Granroth-Wilding, M.; Agres, K.; and Toivonen, H. 2016. Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*. Paris, France: Sony CSL.