Unsupervised Metaphor Identification Using Hierarchical Graph Factorization Clustering

Ekaterina Shutova
International Computer Science Institute and
Institute for Cognitive and Brain Sciences
University of California, Berkeley
katia@icsi.berkeley.edu

Lin Sun
Computer Laboratory
University of Cambridge
lin.sun@cl.cam.ac.uk

Abstract

We present a novel approach to automatic metaphor identification, that discovers both metaphorical associations and metaphorical expressions in unrestricted text. Our system first performs hierarchical graph factorization clustering (HGFC) of nouns and then searches the resulting graph for metaphorical connections between concepts. It then makes use of the salient features of the metaphorically connected clusters to identify the actual metaphorical expressions. In contrast to previous work, our method is fully unsupervised. Despite this fact, it operates with an encouraging precision (0.69) and recall (0.61). Our approach is also the first one in NLP to exploit the cognitive findings on the differences in organisation of abstract and concrete concepts in the human brain.

1 Introduction

Metaphor has traditionally been viewed as a form of linguistic creativity, that gives our expression more vividness, distinction and artistism. While this is true on the surface, the mechanisms of metaphor have a much deeper origin in our reasoning. Today metaphor is widely understood as a cognitive phenomenon operating at the level of mental processes, whereby one concept or domain is systematically viewed in terms of the properties of another (Lakoff and Johnson, 1980). Consider the examples (1) "He *shot down* all of my arguments" and (2) "He *attacked* every weak point in my argument". They demonstrate a metaphorical mapping of the concept of *argument* to that of *war*. The *argument*, which is the target concept, is viewed in terms of a *battle* (or a *war*), the source concept. The existence of such a link allows us to systematically describe *arguments* using the *war* terminology, thus leading to a number of metaphorical expressions. Lakoff and Johnson call such generalisations a source–target domain mapping, or *conceptual metaphor*.

The ubiquity of metaphor in language has been established in a number of corpus studies (Cameron, 2003; Martin, 2006; Steen et al., 2010; Shutova and Teufel, 2010) and the role it plays in human reasoning has been confirmed in psychological experiments (Thibodeau and Boroditsky, 2011). This makes metaphor an important research area for computational and cognitive linguistics, and its automatic processing indispensable for any semantics-oriented NLP application. The problem of metaphor modeling is gaining interest within NLP, with a growing number of approaches exploiting statistical techniques (Mason, 2004; Gedigian et al., 2006; Shutova, 2010; Shutova et al., 2010; Turney et al., 2011; Shutova et al., 2012). Compared to more traditional approaches based on hand-coded knowledge (Fass, 1991; Martin, 1990; Narayanan, 1997; Narayanan, 1999; Feldman and Narayanan, 2004; Barnden and Lee, 2002; Agerri et al., 2007), these more recent methods tend to have a wider coverage, as well as be more efficient, accurate and robust. However, even the statistical metaphor processing approaches so far often focused on a limited domain or a subset of phenomena (Gedigian et al., 2006; Krishnakumaran and Zhu, 2007), and only addressed one of the metaphor processing subtasks: identification of metaphorical mappings (Mason, 2004) or identification of metaphorical expressions (Shutova et al., 2010; Turney et al., 2011). In this paper, we present the first computational method

Proceedings of NAACL-HLT 2013, pages 978–988,
Atlanta, Georgia, 9–14 June 2013.

that identifies the generalisations that govern the production of metaphorical expressions, i.e. conceptual metaphors, and then uses these generalisations to identify metaphorical expressions in unrestricted text. As opposed to previous works on statistical metaphor processing that were supervised or semi-supervised, and thus required training data, our method is fully unsupervised. It relies on building a hierarchical graph of concepts connected by their association strength (using hierarchical clustering) and then searching for metaphorical links in this graph.

Shutova et al. (2010) introduced the hypothesis of "clustering by association" and claimed that in the course of distributional noun clustering, abstract concepts tend to cluster together if they are associated with the same source domain, while concrete concepts cluster by meaning similarity. We share this intuition, but take this idea a significant step further. Our approach is theoretically grounded in the cognitive science findings suggesting that abstract and concrete concepts are organised differently in the human brain (Crutch and Warrington, 2005; Binder et al., 2005; Wiemer-Hastings and Xu, 2005; Huang et al., 2010; Crutch and Warrington, 2010; Adorni and Proverbio, 2012). According to Crutch and Warrington (2005), these differences emerge from their general patterns of relation with other concepts. However, most NLP systems to date treat abstract and concrete concepts as identical. In contrast, we incorporate this distinction into our model by creating a network (or a graph) of concepts, and automatically learning the different patterns of association of abstract and concrete concepts with other concepts. We expect that, while concrete concepts would tend to naturally organise into a tree-like structure (with more specific terms descending from the more general terms), abstract concepts would exhibit a more complex pattern of associations. Consider the example in Figure 1. The figure schematically shows a small portion of the graph describing the concepts of *mechanism* (concrete), *political system* and *relationship* (abstract) at two levels of generality. One can see from this graph that if concrete concepts, such as *bike* or *engine* tend to be connected to only one concept at the higher level in the hierarchy (*mechanism*), abstract concepts may have multiple higher-level associates: the literal ones and the metaphorical ones. For example, the abstract concept of *democracy* is literally associated with a more general concept of *political system*, as well as metaphorically associated with the concept of *mechanism*. Such multiple associations are due to the fact that *political systems* are metaphorically viewed as *mechanisms*, they can *function, break*, they can be *oiled* etc. We often discuss them using *mechanism* terminology, and thus a corpus-based distributional learning approach would learn that they share features with *political systems* (from their literal uses), as well as with *mechanisms* (from their metaphorical uses, as shown next to the respective graph edges in the figure). Our system discovers such association patterns within the graph and uses them to identify metaphorical connections between the concepts.

To the best of our knowledge, our method is the first one to use a hierarchical clustering model for the metaphor processing task. The original graph of concepts is built using hierarchical graph factorization clustering (HGFC) (Yu et al., 2006) of nouns, yielding a network of clusters with different levels of generality. The weights on the edges of the graph indicate association between the clusters (concepts). HGFC has not been previously employed for noun clustering in NLP, but showed successful results in the verb clustering task (Sun and Korhonen, 2011).

In summary, our system (1) builds a graph of concepts using HGFC, (2) traverses it to find metaphorical associations between clusters using weights on the edges of the graph, (3) generates lists of salient features for the metaphorically connected clusters and (4) searches the British National Corpus (BNC) (Burnard, 2007) for metaphorical expressions describing the target domain concepts using the verbs from the set of salient features. We evaluated the performance of the system with the aid of human judges in precision- and recall-oriented settings. In addition, we compared its performance to that of two baselines, an unsupervised baseline using agglomerative clustering (AGG) and a supervised baseline built upon WordNet (Fellbaum, 1998) (WN).

2 Method

2.1 Dataset and Feature Extraction

Our noun dataset used for clustering contains 2000 most frequent nouns in the BNC (Burnard, 2007).

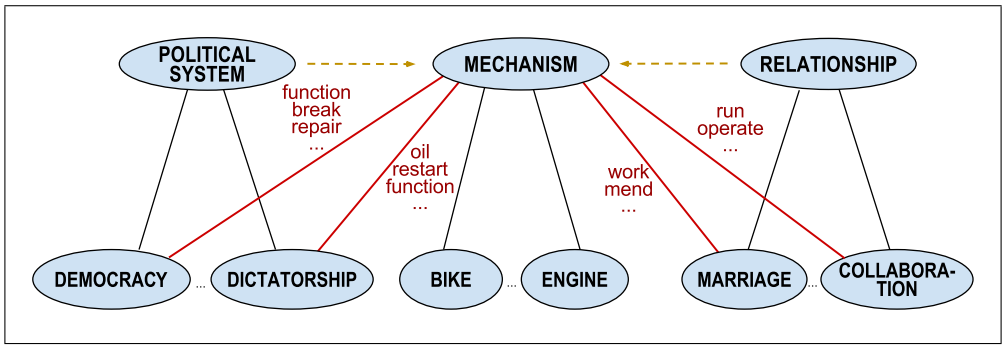


Figure 1: Organisation of the hierarchical graph of concepts

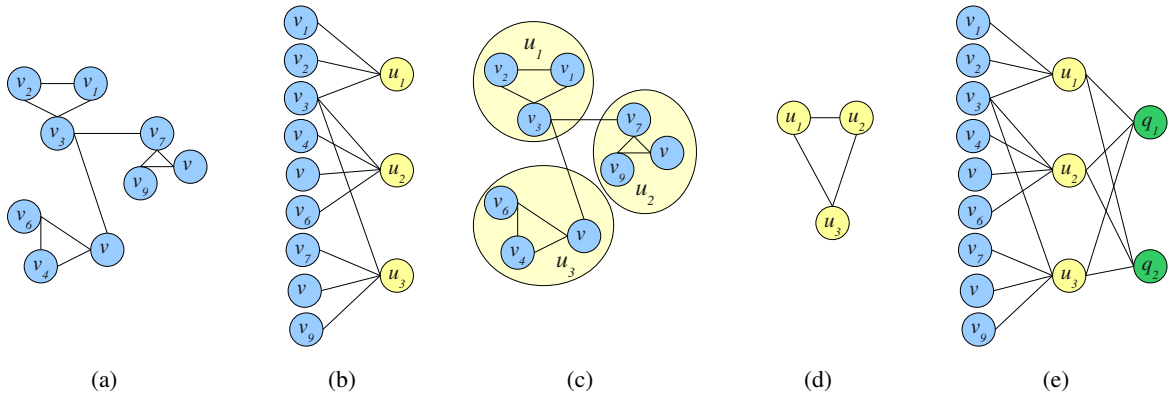
Following previous semantic noun classification experiments (Pantel and Lin, 2002; Bergsma et al., 2008), we use the grammatical relations (GRs) as features for clustering. We extracted the features from the Gigaword corpus (Graff et al., 2003), which was first parsed using the RASP parser (Briscoe et al., 2006). The verb lemmas in VERB–SUBJECT, VERB–DIRECT_OBJECT and VERB–INDIRECT_OBJECT relations with the nouns in the dataset were then extracted from the GR output of the parser. The feature values were the relative frequencies of the features.

2.2 Hierarchical Graph Factorization Clustering

The most widely used method for hierarchical word clustering is AGG (Schulte im Walde and Brew, 2001; Stevenson and Joanis, 2003; Ferrer, 2004; Devereux and Costello, 2005). The method treats each word as a singleton cluster and then successively merges two closest clusters until all the clusters have been merged into one. The cluster similarity is measured using linkage criteria (e.g. Ward (1963) measures the decrease in variance for the clusters being merged). As opposed to this, HGFC derives probabilistic bipartite graphs from the similarity matrix (Yu et al., 2006). Since we require a graph of concepts, our task is rather different from standard hierarchical word clustering that produces a tree of concepts. In a tree, each word can only have a unique parent cluster at each level. Our concept graph does not have this constraint: at any level a word can be associated with an arbitrary number of parent clusters. Therefore, not only HGFC outperformed agglomerative clustering methods in hierarchical clustering tasks (Yu et al., 2006; Sun and Korhonen, 2011), but its hierarchical graph output is also a more suitable representation of the concept graph. In addition, HGFC can detect the number of levels and the number of clusters on each level of the hierarchical graph automatically. This is essential for our task as these settings are difficult to predefine for a general-purpose concept graph.

Given a set of nouns, $V = \{v_n\}_{n=1}^{N}$, the similarity matrix W for HGFC is constructed using Jensen-Shannon Divergence. W can be encoded by an undirected graph G (Figure 2(a)), where the nouns are mapped to vertices and W_{ij} is the edge weight between vertices i and j. The graph G and the cluster structure can be represented by a bipartite graph $K(V, U)$. V are the vertices on G. $U = \{u_p\}_{p=1}^{m}$ represent the hidden m clusters. For example, looking at Figure 2(b), V on G can be grouped into three clusters u_1, u_2 and u_3. The matrix B denotes the $n \times m$ adjacency matrix, with b_{ip} being the connection weight between the vertex v_i and the cluster u_p. Thus, B represents the connections between clusters at an upper and lower level of clustering. A flat clustering algorithm can be induced by assigning a lower level node to the parent node that has the largest connection weight. The number of clusters at any level can be determined by only counting the number of non-empty nodes (namely the nodes that have at least one lower level node associated).

The bipartite graph K also induces a similarity (W') between v_i and v_j: $w'_{ij} = \sum_{p=1}^{m} \frac{b_{ip} b_{jp}}{\lambda_p} = (B\Lambda^{-1}B^T)_{ij}$ where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_m)$. Therefore, B can be found by minimizing the divergence distance (ζ) between the similarity matrices W and W':



(a) (b) (c) (d) (e)

Figure 2: (a) An undirected graph G representing the similarity matrix; (b) The bipartite graph showing three clusters on G; (c) The induced clusters U; (d) The new graph G_1 over clusters U; (e) The new bipartite graph over G_1

$$\min_{H,\Lambda} \zeta(W, H\Lambda H^T), s.t. \sum_{i=1}^{n} h_{ip} = 1 \quad (1)$$

$$H = B\Lambda^{-1}; \zeta(X,Y) = \sum_{ij}(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij})$$

Yu et al. (2006) showed that this cost function is non-increasing under the update rule:

$$\tilde{h}_{ip} \propto h_{ip} \sum_j \frac{w_{ij}}{(H\Lambda H^T)_{ij}} \lambda_p h_{jp} \text{ s.t.} \sum_i \tilde{h}_{ip} = 1 \quad (2)$$

$$\tilde{\lambda}_p \propto \lambda_p \sum_j \frac{w_{ij}}{(H\Lambda H^T)_{ij}} h_{ip} h_{jp} \text{ s.t.} \sum_p \tilde{\lambda}_p = \sum_{ij} w_{ij} \quad (3)$$

The cost function can thus be optimized by updating h and λ alternately.

The similarity between clusters $p(u_p, u_q)$ can be induced from B, as follows:

$$p(u_p, u_q) = p(u_p)p(u_p|u_q) = (B^T D^{-1} B)_{pq} \quad (4)$$

$$D = \text{diag}(d_1, ..., d_n) \text{ where } d_i = \sum_{p=0}^{m} b_{ip}$$

We can then construct a new graph G_1 (Figure 2(d)) with the clusters U as vertices, and the cluster similarity p as the connection weight. The clustering algorithm can now be applied again (Figure 2(e)). This process can go on iteratively, leading to a hierarchical graph.

The number of levels (L) and the number of clusters (m_l) are detected automatically, using the method of Sun and Korhonen (2011). Clustering starts with an initial setting of number of clusters (m_0) for the first level. In our experiment, we set the value of m_0 to 800. For the subsequent levels, m_l is set to the number of non-empty clusters (bipartite graph nodes) on the parent level. The matrices B and Λ are initialized randomly. We found that the actual initialization values have little impact on the final result. The rows in B are normalized after the initialization so the values in each row add up to one. For a word v_i, the probability of assigning it to cluster $x_p^{(l)} \in X_l$ at level l is given by:

$$p(x_p^{(l)}|v_i) = \sum_{X_{l-1}} ... \sum_{x^{(1)} \in X_1} p(x_p^{(l)}|x^{(l-1)})...p(x^{(1)}|v_i)$$

$$= (D_1^{(-1)} B_1 D_2^{-1} B_2 ... D_l^{-1} B_l)_{ip} \quad (5)$$

Due to the random walk property of the graph, m_l is non-increasing for higher levels (Sun and Korhonen, 2011). The algorithm can thus terminate when all nouns are assigned to one cluster. We run 1000 iterations of updates of h and λ (equation 2 and 3) for each two adjacent levels.

The resulting graph is composed of a set of bipartite graphs defined by $B_l, B_{l-1}, ..., B_1$. A bipartite graph has a similar structure as in Figure 1. For a given noun, we can rank the clusters at any level according to the soft assignment probability (eq. 5). The clusters that have no member noun were hidden from the ranking since they do not explicitly represent any concept. However, these clusters are still part of the organisation of conceptual space within the model and they contribute to the probability for the clusters on upper levels (eq. 5). We call the *view* of the hierarchical graph where these empty clusters

are hidden an *explicit graph*. The whole algorithm can be summarized as follows:

Require: N nouns V, initial number of clusters m_1
  Compute the similarity matrix W_0 from V
  Build the graph G_0 from W_0, $l \leftarrow 1$
  while $m_l > 1$ **do**
    Factorize G_{l-1} to obtain bipartite graph K_l with the adjacency matrix B_l (eq. 1, 2 and 3)
    Build a graph G_l with similarity matrix $W_l = B_l^T D_l^{-1} B_l$ according to equation 4
    $l \leftarrow l + 1$; $m_l \leftarrow$ No. non-empty clusters (eq. 5)
  end while
  return $B_l, B_{l-1}...B_1$

2.3 Identification of Metaphorical Associations

Once we obtained the explicit graph of concepts, we can now identify metaphorical associations based on the weights connecting the clusters at different levels (eq. 5). Taking a single noun (e.g. *fire*) as input, the system computes the probability of its cluster membership for each cluster at each level, using these weights. We expect the cluster membership probabilities to indicate the level of association of the input noun with the clusters. The system can then rank the clusters at each level based on these probabilities. We chose level 3 as the optimal level of generality for our experiments, based on our qualitative analysis of the graph. The system selects 6 top-ranked clusters from this level (we expect an average source concept to have no more than 5 typical target associates) and excludes the literal cluster containing the input concept (e.g. "*fire flame blaze*"). The remaining clusters represent the target concepts associated with the input source concept. Example output for the input concepts of *fire* and *disease* is shown in Figure 3. One can see from the Figure that each of the noun-to-cluster mappings represents a new conceptual metaphor, e.g. EMOTION is FIRE, VIOLENCE is FIRE, CRIME is a DISEASE etc. These mappings are exemplified in language by a number of metaphorical expressions (e.g. "His anger will *burn* him", "violence *flared* again", "it's time they found a *cure* for corruption").

2.4 Identification of Salient Features and Metaphorical Expressions

After extracting the source–target domain mappings, we now move on to the identification of the corresponding metaphorical expressions. The system does this by harvesting the salient features that lead to the input noun being strongly associated with the extracted clusters. The salient features are selected by ranking the features according to the joint probability of the feature (f) occurring both with the input noun (w) and the cluster (c). Under a simplified independence assumption, $p(w, c|f) = p(w|f) \times p(c|f)$. $p(w|f)$ and $p(c|f)$ are calculated as the ratio of the frequency of the feature f to the total frequency of the input noun and the cluster respectively. The features ranked higher are expected to represent the source domain vocabulary that can be used to metaphorically describe the target concepts. We selected the top 50 features from the ranked list. Example features (verbs and their grammatical relations) extracted for the source domain noun *fire* and the *violence* cluster are shown in Figure 4.

We then refined the lists of features by means of selectional preference (SP) filtering. We use SPs to

SOURCE: fire
TARGET 1: sense hatred emotion passion enthusiasm sentiment hope interest feeling resentment optimism hostility excitement anger
TARGET 2: coup violence fight resistance clash rebellion battle drive fighting riot revolt war confrontation volcano row revolution struggle
TARGET 3: alien immigrant
TARGET 4: prisoner hostage inmate

SOURCE: disease
TARGET 1: fraud outbreak offense connection leak count crime violation abuse conspiracy corruption terrorism suicide
TARGET 2: opponent critic rival
TARGET 3: execution destruction signing
TARGET 4: refusal absence fact failure lack delay

Figure 3: Discovered metaphorical associations

rage-ncsubj *engulf*-ncsubj *erupt*-ncsubj *burn*-ncsubj *light*-dobj *consume*-ncsubj *flare*-ncsubj *sweep*-ncsubj *spark*-dobj *battle*-dobj *gut*-idobj *smolder*-ncsubj *ignite*-dobj *destroy*-idobj *spread*-ncsubj *damage*-idobj *light*-ncsubj *ravage*-ncsubj *crackle*-ncsubj *open*-dobj *fuel*-dobj *spray*-idobj *roar*-ncsubj *perish*-idobj *destroy*-ncsubj *wound*-idobj *start*-dobj *ignite*-ncsubj *injure*-idobj *fight*-dobj *rock*-ncsubj *retaliate*-idobj *devastate*-idobj *blaze*-ncsubj *ravage*-idobj *rip*-ncsubj *burn*-idobj *spark*-ncsubj *warm*-idobj *suppress*-dobj *rekindle*-dobj

Figure 4: Salient features for *fire* and the *violence* cluster

FEELING IS FIRE
hope *lit* (Subj), anger *blazed* (Subj), optimism *raged* (Subj), enthusiasm *engulfed* them (Subj), hatred *flared* (Subj), passion *flared* (Subj), interest *lit* (Subj), *fuel* resentment (Dobj), anger *crackled* (Subj), feelings *roared* (Subj), hostility *blazed* (Subj), *light* with hope (Iobj)

CRIME IS A DISEASE
cure crime (Dobj), abuse *transmitted* (Subj), *eradicate* terrorism (Dobj), *suffer from* corruption (Iobj), *diagnose* abuse (Dobj), *combat* fraud (Dobj), *cope with* crime (Iobj), *cure* abuse (Dobj), *eradicate* corruption

Figure 5: Identified metaphorical expressions for the mappings FEELING IS FIRE and CRIME IS A DISEASE

quantify how well the extracted features describe the source domain (e.g. *fire*). We extracted nominal argument distributions of the verbs in our feature lists for VERB–SUBJECT, VERB–DIRECT_OBJECT and VERB–INDIRECT_OBJECT relations. We used the algorithm of Sun and Korhonen (2009) to create SP classes and the measure of Resnik (1993) to quantify how well a particular argument class fits the verb. Resnik measures selectional preference strength $S_R(v)$ of a predicate as a Kullback-Leibler distance between two distributions: the prior probability of the noun class $P(c)$ and the posterior probability of the noun class given the verb $P(c|v)$. $S_R(v) = D(P(c|v)||P(c)) = \sum_c P(c|v) \log \frac{P(c|v)}{P(c)}$. In order to quantify how well a particular argument class fits the verb, Resnik defines selectional association as $A_R(v,c) = \frac{1}{S_R(v)} P(c|v) \log \frac{P(c|v)}{P(c)}$. We rank the nominal arguments of the verbs in our feature lists using their selectional association with the verb, and then only retain the features whose top 5 arguments contain the source concept. For example, the verb *start*, that is a common feature for both *fire* and the *violence* cluster (e.g. "start a war", "start a fire"), would be filtered out in this way, whereas the verbs *flare* or *blaze* would be retained as descriptive source domain vocabulary.

We then search the RASP-parsed BNC for grammatical relations, in which the nouns from the target domain cluster appear with the verbs from the source domain vocabulary (e.g. "war *blazed*" (subj), "to *fuel* violence" (dobj) for the mapping VIOLENCE is FIRE). The system thus annotates metaphorical expressions in text, as well as the corresponding conceptual metaphors, as shown in Figure 5.

3 Evaluation and Discussion

3.1 Baselines

AGG: the agglomerative clustering baseline is constructed using SciPy implementation (Oliphant, 2007) of Ward's linkage method (Ward, 1963). The output tree is cut according to the number of levels and the number of clusters of the explicit graph detected by HGFC. The resulting tree is converted into a graph by adding connections from each cluster to all the clusters one level above. The connection weight (the cluster distance) is measured using Jensen-Shannon Divergence between the cluster centroids. This graph is used in place of the HGFC graph in the metaphor identification experiments.
WN: in the WN baseline, the WordNet hierarchy is used as the underlying graph of concepts, to which the metaphor extraction method is applied. Given a source concept, the system extracts all its sense-1 hypernyms two levels above and subsequently all of their sister terms. The hypernyms themselves are considered to represent the literal sense of the source noun and are, therefore, removed. The sister terms are kept as potential target domains.

3.2 Evaluation of Metaphorical Associations

To create our dataset, we extracted 10 common source concepts that map to multiple targets from the Master Metaphor List (Lakoff et al., 1991) and linguistic analyses of metaphor (Lakoff and Johnson, 1980; Shutova and Teufel, 2010). These included FIRE, CHILD, SPEED, WAR, DISEASE, BREAKDOWN, CONSTRUCTION, VEHICLE, SYSTEM, BUSINESS. Each of the three systems identified 50 source–target domain mappings for the given source domains, resulting in a set of 150 conceptual metaphors (each representing a number of submappings since all the target concepts are clusters or synsets). These were then evaluated against human judgements in two different experimental settings.
Setting 1: The judges were presented with a set of conceptual metaphors identified by the three systems, randomized. They were asked to annotate the mappings they considered valid. In all our experiments, the judges were encouraged to rely on their own intuition of metaphor, but they also reviewed the metaphor annotation guidelines of Shutova and Teufel (2010). Two independent judges, both na-

tive speakers of English, participated in this experiment. Their agreement on the task was $\kappa = 0.60$ ($n = 2, N = 150, k = 2$) (Siegel and Castellan, 1988). The main differences in the annotators' judgements stem from the fact that some metaphorical associations are less obvious and common than others, and thus need more context (or imaginative effort) to establish. Such examples, where the judges disagreed included metaphorical mappings such as INTENSITY is SPEED, GOAL is a CHILD, COLLECTION is a SYSTEM, ILLNESS is a BREAKDOWN.

The system performance was then evaluated against these judgements in terms of precision (P), i.e. the proportion of the valid metaphorical mappings among those identified. We calculated system precision (in all experiments) as an average over both annotations. HGFC operates with a precision of $P = 0.69$, whereas the baselines attain $P = 0.36$ (AGG) and $P = 0.29$ (WN). The precision of annotator judgements against each other (the human ceiling) is $P = 0.80$, suggesting that this is a challenging task.

Setting 2: To measure recall, R, of the systems we asked two annotators (both native speakers with a background in metaphor, different from Setting 1) to write down up to 5 target concepts they strongly associated with each of the 10 source concepts. Their annotations were then aggregated into a single metaphor association gold standard, consisting of 63 mappings in total. The recall of the systems was measured against this gold standard, resulting in HGFC $R = 0.61$, AGG $R = 0.11$ and WN $R = 0.03$.

As expected, HGFC outperforms both AGG and WN baselines in both settings. AGG has been previously shown to be less accurate than HGFC in the verb clustering task (Sun and Korhonen, 2011). Our analysis of the noun clusters indicated that HGFC tends to produce more pure and complete clusters than AGG. Another important reason AGG fails is that it by definition organises all concepts into tree and optimises its solution locally, taking into account a small number of clusters at a time. However, being able to discover connections between more distant domains and optimising globally over all concepts is crucial for metaphor identification. This makes AGG less suitable for the task, as demonstrated by our results. However, AGG identified a number of interesting mappings missed by HGFC, e.g. CAREER IS A CHILD, LANGUAGE IS A SYSTEM, CORRUPTION IS A VEHICLE, EMPIRE IS A CONSTRUCTION, as well as a number of mappings in common with HGFC, e.g. DEBATE IS A WAR, DESTRUCTION IS A DISEASE. The WN system also identified a few interesting metaphorical mappings (e.g. COGNITION IS FIRE, EDUCATION IS CONSTRUCTION), but its output is largely dominated by the concepts similar to the source noun and contains some unrelated concepts. The comparison of HGFC to WN shows that HGFC identifies meaningful properties and relations of abstract concepts that can not be captured in a tree-like classification (even an accurate, manually created one). The latter is more appropriate for concrete concepts, and a more flexible representation is needed to model abstract concepts. The fact that both baselines identified some valid metaphorical associations, relying on less suitable conceptual graphs, suggests that our way of traversing the graph is a viable approach in principle.

HGFC identifies valid metaphorical associations for a range of source concepts. On of them (CRIME IS A VIRUS) happened to have been already validated in psychological experiments (Thibodeau and Boroditsky, 2011). The most frequent type of error of HGFC is the presence of target clusters similar or closely related to the source noun (e.g. the *parent* cluster for *child*). The clusters from the same domain can, however, be filtered out if their nouns frequently occur in the same documents with the source noun (in a large corpus), i.e. by topical similarity. The latter is less likely for the metaphorically connected nouns. We intend to implement this improvement in the future version of the system.

3.3 Evaluation of Metaphorical Expressions

For each of the identified conceptual metaphors, the three systems extracted a number of metaphorical expressions from the corpus (average of 430 for HGFC, 148 for AGG, and 855 for WN). The expressions were also evaluated against human judgements. The judges were presented with a set of randomly sampled sentences containing metaphorical expressions as annotated by the system and by the baselines (200 each), randomized. They were asked to mark the tagged expressions that were metaphorical in their judgement as correct. Their agreement on the task was $\kappa = 0.56$ ($n = 2, N = 600, k = 2$),

HLJ 26 [..] "effective action" was needed to **eradicate terrorism, drug-trafficking** and **corruption**.
EG0 275 In the 1930s the words "means test" was a curse, **fuelling the resistance** against it both among the unemployed and some of its administrators.
CRX 1054 [..] if the rehabilitative approach were demonstrably successful in **curing crime**.
HL3 1206 [..] he would strive to **accelerate progress** towards the economic integration of the Caribbean.
HXJ 121 [..] it is likely that some **industries will flourish** in certain countries as the **market widens**.

Figure 6: Metaphors tagged by the system (in bold)

whereby the main source of disagreement was the presence of lexicalized metaphors, e.g. verbs such as *impose, decline* etc. The system performance against these annotations is $P = 0.65$ (HGFC), $P = 0.47$ (AGG) and $P = 0.12$ (WN). The human ceiling for this task was measured at $P = 0.79$. Figure 6 shows example sentences annotated by HGFC. The performance of our unsupervised approach is close to the previous supervised systems of Mason (2004) (accuracy of 0.73) and Shutova et al. (2010) (precision of 0.79), however, the results are not directly comparable due to different experimental settings.

The system errors in this task stem from multiple word senses of the salient features or the source and target sharing some physical properties (e.g. one can "*die from* crime" and "*die from* a disease"). Some identified expressions invoke a chain of mappings (e.g. ABUSE IS A DISEASE, DISEASE IS AN ENEMY for "*combat* abuse"), however, such chains are not yet incorporated into the system. The performance of AGG is higher than in the mappings identification task, since it outputs only few expressions for the incorrect mappings. In contrast, WN tagged a large number of literal expressions due to the incorrect prior identification of the underlying associations.

Since there is no large metaphor-annotated corpus available, it was impossible for us to reliably evaluate the recall of metaphorical expressions. However, we estimated it as a recall of salient features. We manually compiled sets of typical features for the 10 source domains, and measured their recall among the top 50 HGFC features at $R = 0.70$. However, in practice the coverage in this task would directly depend on that of the metaphorical associations.

4 Related Work

One of the first attempts to identify and interpret metaphorical expressions in text is the met* system of Fass (1991), that utilizes hand-coded knowledge and detects non-literalness via selectional preference violation. In case of a violation, the respective phrase is first tested for being metonymic using hand-coded patterns (e.g. CONTAINER-FOR-CONTENT). If this fails, the system searches the knowledge base for a relevant analogy in order to discriminate metaphorical relations from anomalous ones. The system of Krishnakumaran and Zhu (2007) uses WordNet (the hyponymy relation) and word bigram counts to predict verbal, nominal and adjectival metaphors at the sentence level. The authors discriminate between conventional metaphors (included in WordNet) and novel metaphors. Birke and Sarkar (2006) present a sentence clustering approach that employs a set of seed sentences annotated for literalness and computes similarity between the new input sentence and all of the seed sentences. The system then tags the sentence as literal or metaphorical according to the annotation in the most similar seeds, attaining an f-score of 53.8%.

The first system to discover source–target domain mappings automatically is CorMet (Mason, 2004). It does this by searching for systematic variations in domain-specific verb selectional preferences. For example, *pour* is a characteristic verb in both LAB and FINANCE domains. In the LAB domain it has a strong preference for *liquids* and in the FINANCE domain for *money*. From this the system infers the domain mapping FINANCE – LAB and the concept mapping *money* – *liquid*. Gedigian et al. (2006) trained a maximum entropy classifier to discriminate between literal and metaphorical use. They annotated the sentences from PropBank (Kingsbury and Palmer, 2002) containing the verbs of MOTION and CURE for metaphoricity. They used PropBank annotation (arguments and their semantic types) as features for classification and report an accuracy of 95.12% (however, against a majority baseline of 92.90%). The metaphor identification system of Shutova et al. (2010) starts from a small seed set of metaphorical expressions, learns the analogies involved in their production and extends the set of analogies by means of verb and noun clustering. As

a result, the system can recognize new metaphorical expressions in unrestricted text (e.g. from the seed "*stir* excitement" it infers that "*swallow* anger" is also a metaphor), achieving a precision of 79%.

Turney et al. (2011) classify verbs and adjectives as literal or metaphorical based on their level of concreteness or abstractness in relation to a noun they appear with. They learn concreteness rankings for words automatically (starting from a set of examples) and then search for expressions where a concrete adjective or verb is used with an abstract noun (e.g. "*dark* humour" is tagged as a metaphor and "dark hair" is not). They report an accuracy of 73%.

5 Conclusions and Future Directions

Previous research on metaphor addressed a number of different aspects of the phenomenon, and has shown that these aspects can be successfully modeled using statistical techniques. However, the methods often focused on a limited domain and needed manually-labeled training data. This made them difficult to apply in a real-world setting with the goal of improving semantic interpretation in NLP at large. Our method takes a step towards this direction. It is fully unsupervised, and thus more robust, and can perform accurate metaphor identification in unrestricted text. It identifies metaphor with a precision of 69% and a recall of 61%, which is a very encouraging result for an unsupervised method. We believe that this work has important implications for computational and cognitive modeling of metaphor, but is also applicable to a range of other semantic tasks within NLP. Integrating different representations of abstract and concrete concepts into NLP systems may improve their performance, as well as make the models more cognitively plausible.

One of our key future research objectives is to investigate the use and adaptation of the created conceptual graph to perform metaphor interpretation. In addition, we plan to extend this work to cover nominal and adjectival metaphors, by harvesting salient nominal and adjectival features.

Acknowledgments

This work was funded by the MetaNet project (grant number W911NF-12-C-0022) and the Dorothy Hodgkin Postgraduate Award.

References

Roberta Adorni and Alice Mado Proverbio. 2012. The neural manifestation of the word concreteness effect: An electrical neuroimaging study. *Neuropsychologia*, 50(5):880 – 891.

Rodrigo Agerri, John Barnden, Mark Lee, and Alan Wallington. 2007. Metaphor, inference and domain-independent mappings. In *Proceedings of RANLP-2007*, pages 17–23, Borovets, Bulgaria.

John Barnden and Mark Lee. 2002. An artificial intelligence approach to metaphor understanding. *Theoria et Historia Scientiarum*, 6(1):399–412.

Shane Bergsma, Dekang Lin, and Randy Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 59–68, Honolulu, Hawaii.

Jeffrey R. Binder, Chris F. Westbury, Kristen A. McKiernan, Edward T. Possing, and David A. Medler. 2005. Distinct brain systems for processing concrete and abstract concepts. *Journal of Cognitive Neuroscience*, 17(6):905–917.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for the nearly unsupervised recognition of nonliteral language. In *In Proceedings of EACL-06*, pages 329–336.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the rasp system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*.

Lou Burnard. 2007. *Reference Guide for the British National Corpus (XML Edition)*.

Lynne Cameron. 2003. *Metaphor in Educational Discourse*. Continuum, London.

Sebastian J. Crutch and Elizabeth K. Warrington. 2005. Abstract and concrete concepts have structurally different representational frameworks. *Brain*, 128(3):615–627.

Sebastian J Crutch and Elizabeth K Warrington. 2010. The differential dependence of abstract and concrete words upon associative and similarity-based information: Complementary semantic interference and facilitation effects. *Cognitive Neuropsychology*, 27(1):46–71.

Barry Devereux and Fintan Costello. 2005. Propane stoves and gas lamps: How the concept hierarchy influences the interpretation of noun-noun compounds. In *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society*.

Dan Fass. 1991. met*: A method for discriminating metonymy and metaphor by computer. *Computational Linguistics*, 17(1):49–90.

Jerome Feldman and Srini Narayanan. 2004. Embodied meaning in a neural theory of language. *Brain and Language*, 89(2):385–392.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database (ISBN: 0-262-06197-X)*. MIT Press, first edition.

Eva E. Ferrer. 2004. Towards a semantic classification of spanish verbs based on subcategorisation information. In *Proceedings of the ACL 2004 workshop on Student research*, page 13. Association for Computational Linguistics.

Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ciric. 2006. Catching metaphors. In *In Proceedings of the 3rd Workshop on Scalable Natural Language Understanding*, pages 41–48, New York.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

Hsu-Wen Huang, Chia-Lin Lee, and Kara D. Federmeier. 2010. Imagine that! erps provide evidence for distinct hemispheric contributions to the processing of concrete and abstract concepts. *NeuroImage*, 49(1):1116 – 1123.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of LREC-2002*, pages 1989–1993, Gran Canaria, Canary Islands, Spain.

Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting elusive metaphors using lexical resources. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 13–20, Rochester, NY.

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

George Lakoff, Jane Espenson, and Alan Schwartz. 1991. The master metaphor list. Technical report, University of California at Berkeley.

James Martin. 1990. *A Computational Model of Metaphor Interpretation*. Academic Press Professional, Inc., San Diego, CA, USA.

James Martin. 2006. A corpus-based analysis of context effects on metaphor comprehension. In A. Stefanowitsch and S. T. Gries, editors, *Corpus-Based Approaches to Metaphor and Metonymy*, Berlin. Mouton de Gruyter.

Zachary Mason. 2004. Cormet: a computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, 30(1):23–44.

Srini Narayanan. 1997. Knowledge-based Action Representations for Metaphor and Aspect (KARMA). Technical report, PhD thesis, University of California at Berkeley.

Srini Narayanan. 1999. Moving right along: A computational model of metaphoric reasoning about events. In *Proceedings of AAAI 99)*, pages 121–128, Orlando, Florida.

Travis E. Oliphant. 2007. Python for scientific computing. *Computing in Science and Engineering*, 9:10–20.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM.

Philip Resnik. 1993. *Selection and Information: A Class-based Approach to Lexical Relationships*. Ph.D. thesis, Philadelphia, PA, USA.

Sabine Schulte im Walde and Chris Brew. 2001. Inducing German semantic verb classes from purely syntactic subcategorisation information. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 223–230, Morristown, NJ, USA. Association for Computational Linguistics.

Ekaterina Shutova and Simone Teufel. 2010. Metaphor corpus annotated for source - target domain mappings. In *Proceedings of LREC 2010*, pages 3255–3261, Malta.

Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of Coling 2010*, pages 1002–1010, Beijing, China.

Ekaterina Shutova, Simone Teufel, and Anna Korhonen. 2012. Statistical Metaphor Processing. *Computational Linguistics*, 39(2).

Ekaterina Shutova. 2010. Automatic metaphor interpretation as a paraphrasing task. In *Proceedings of NAACL 2010*, pages 1029–1037, Los Angeles, USA.

Sidney Siegel and N. John Castellan. 1988. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill Book Company, New York, USA.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna A. Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*. John Benjamins, Amsterdam/Philadelphia.

Suzanne Stevenson and Eric Joanis. 2003. Semi-supervised verb class discovery using noisy features. In *Proceedings of HLT-NAACL 2003*, pages 71–78.

Lin Sun and Anna Korhonen. 2009. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of EMNLP 2009*, pages 638–647, Singapore, August.

Lin Sun and Anna Korhonen. 2011. Hierarchical verb clustering using graph factorization. In *Proceedings of EMNLP*, pages 1023–1033, Edinburgh, UK.

Paul H. Thibodeau and Lera Boroditsky. 2011. Metaphors we think with: The role of metaphor in reasoning. *PLoS ONE*, 6(2):e16782, 02.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 680–690, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joe H. Ward. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Katja Wiemer-Hastings and Xu Xu. 2005. Content Differences for Abstract and Concrete Concepts. *Cognitive Science*, 29(5):719–736.

Kai Yu, Shipeng Yu, and Volker Tresp. 2006. Soft clustering on graphs. *Advances in Neural Information Processing Systems*, 18.