

HARPY: Hypernyms and Alignment of Relational Paraphrases

Adam Grycner

Max-Planck Institute for Informatics
Campus E1.4, 66123
Saarbrücken, Germany
agrycner@mpi-inf.mpg.de

Gerhard Weikum

Max-Planck Institute for Informatics
Campus E1.4, 66123
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

Abstract

Collections of relational paraphrases have been automatically constructed from large text corpora, as a WordNet counterpart for the realm of binary predicates and their surface forms. However, these resources fall short in their coverage of hypernymy links (subsumptions) among the synsets of phrases. This paper closes this gap by computing a high-quality alignment between the relational phrases of the Patty taxonomy, one of the largest collections of this kind, and the verb senses of WordNet. To this end, we devise judicious features and develop a graph-based alignment algorithm by adapting and extending the SimRank random-walk method. The resulting taxonomy of relational phrases and verb senses, coined HARPY, contains 20,812 synsets organized into a *Directed Acyclic Graph (DAG)* with 616,792 hypernymy links. Our empirical assessment, indicates that the alignment links between Patty and WordNet have high accuracy, with *Mean Reciprocal Rank (MRR)* score 0.7 and *Normalized Discounted Cumulative Gain (NDCG)* score 0.73. As an additional extrinsic value, HARPY provides fine-grained lexical types for the arguments of verb senses in WordNet.

1 Introduction

Motivation: This paper addresses the task of discovering and organizing paraphrases of relations between entities (Lin and Pantel, 2001; Fader et al., 2011; Nakashole et al., 2012; Moro and Navigli, 2012; Alfonseca et al., 2013). This task involves understanding that the phrases “travels to”, “visits” and “on her tour through” (relating a person and a country) are synonymous and that “leader of” and “works with” (relating a person and an organization) are in a hypernymy relation: the former is subsumed by the latter. This kind of lexical knowledge can be harnessed for advanced tasks like question answering (Fader et al., 2013), search over web tables (Gupta et al., 2014), or event mining over news (Alfonseca et al., 2013).

Work along these lines has developed large repositories of relational paraphrases, most notably, the collections ReVerb (Fader et al., 2011), Patty (Nakashole et al., 2012), and WiSeNet (Moro and Navigli, 2012). The largest of these, Patty, contains ca. 350,000 synsets of phrases, each annotated with ontological types of their two arguments (e.g., person \times country, or politician \times political_party). However, the subsumption hierarchy of Patty is very sparse. It contains only 8,000 hypernymy links between phrases, and the entire taxonomy is kind of fragmented into a many-rooted DAG (directed acyclic graph). Moreover, the synsets are rather noisy in the long tail with low confidence. WiSeNet, an alternative resource, has ca. 40,000 synsets and no hypernymy links.

WordNet (Fellbaum, 1998), on the other hand, is a very rich resource on synonymy and hypernymy. However, its coverage of binary relations (as opposed to unary predicates, mostly nouns) is restricted to (mostly) single-word verbs. WordNet has ca. 13,767 verb synsets, organized into a hierarchy with 13,239 hypernymy links. Unlike Patty, though, WordNet does not associate verb senses with a lexical type signature for the subject and object arguments of a verb, and it is sparse in multi-word phrases.

This work is licenced under a Creative Commons Attribution 4.0 International License. Page numbers and proceedings footer are added by the organizers. License details: <http://creativecommons.org/licenses/by/4.0/>

Resources like VerbNet (Kipper et al., 2008) or FrameNet (Baker et al., 1998) aim to overcome these deficiencies, but are much smaller.

Goal and Approach: In this paper, our goal is to overcome the limitations of resources like Patty and WordNet. We want to reconcile the wealth of Patty’s multi-word paraphrases with lexical typing, on one hand, and the clean hypernymy organization of WordNet verbs, on the other hand. To this end, we compute an alignment between the phrase synsets that Patty provides with the verb senses of WordNet. This has mutual benefits: 1) we enhance many Patty phrases with the clean hypernyms of WordNet, this way augmenting the subsumption hierarchy, and 2) we extend WordNet verb senses with the lexical type signatures derived from Patty. Our approach uses a variety of features from both of the two aligned resources, as well as further auxiliary sources. Algorithmically, we build on an advanced notion of random walks over graphs, known as SimRank (Jeh and Widom, 2002).

Contributions: Our method is able to construct a high-quality taxonomy of relational paraphrases, coined *HARPY*, that combines the richness of Patty with the clean hierarchy of WordNet. The algorithm for computing the alignment is efficient and robust. One can think of the alignment as a way of sense-disambiguating Patty phrases by mapping them to WordNet. *HARPY* links 20,812 of the Patty phrases to WordNet. Conversely, 4,789 out of 13,767 WordNet verb senses are enriched with information from Patty. We evaluate the quality of *HARPY* by extensive sampling with human assessment. We also demonstrate its benefit by the extrinsic use-case of annotating WordNet verb senses with lexical type signatures. All experimental data and the *HARPY* resource will be available on a public web site.

2 Related Work

With the proliferation of knowledge bases, like Freebase (Google Knowledge Graph), DBpedia, YAGO, or ConceptNet, there is a wealth of resources about entities and semantic classes (i.e., unary predicates and their instances). In contrast, the systematic compilation of paraphrases for relations (i.e., binary predicates) has received much less attention. Some of the knowledge-base projects, especially those that center on Open Information Extraction, make intensive use of surface patterns (e.g., verbal phrases) that indicate relations (e.g., (Carlson et al., 2010; Fader et al., 2011; Mausam et al., 2012; Speer and Havasi, 2012; Wu et al., 2012)); however, they do not organize these patterns into a WordNet-style taxonomy.

Prior work towards such taxonomies go back to the projects DIRT (Lin and Pantel, 2001), VerbOcean (Chklovski and Pantel, 2004), and VerbNet (Kipper et al., 2008). However, the resulting resources were mostly restricted to single verbs. ReVerb (Fader et al., 2011) extended these approaches by automatically mining entire phrases from Web contents, but still with focus on verbal structures. Patty (Nakashole et al., 2012) used sequence mining algorithms for gathering a general class of relational phrases, organizing them into synsets, and inferring lexical type signatures. WiseNet (Moro and Navigli, 2012) harnessed phrases from Wikipedia articles and clustered them into synsets of relational phrases. All of these works are fairly limited in their coverage of subsumptions (hypernymy) between relational phrases.

There is ample work on computing alignments among different kinds of lexical thesauri, dictionaries, taxonomies, ontologies, and other forms of linguistic or semantic resources. Prominent cases along these lines include the alignments between FrameNet and WordNet (Ferrández et al., 2010), VerbNet and PropBank (Palmer, 2009), Wikionary and WordNet (Meyer and Gurevych, 2012), and across multilingual WordNets and/or Wikipedia editions (e.g., (de Melo and Weikum, 2009; Navigli and Ponzetto, 2012)). For aligning ontologies based on OWL and RDF logics, there is a series of annual benchmark competitions (Grau et al., 2013). Most approaches are based on relatedness measures and context similarities between words or concepts and their neighborhoods in the respective resources (e.g., (Banerjee and Pedersen, 2003; Budanitsky and Hirst, 2006; Gabrilovich and Markovitch, 2007)). Algorithmically, this translates into a nearest-neighbor (most-similar) assignment between entries of different resources. More sophisticated methods use similarities merely to assign weights to relatedness edges in a graph, and then employ random walks on such a graph (e.g., (Pilehvar et al., 2013)). The prevalent method of this kind uses Personalized Page Rank (Haveliwala, 2002)), computing stationary probabilities for reaching nodes in one resource when starting random walks on a given node of the other resources (with randomized restarts).

Computing alignments between resources can sometimes be viewed as a task of disambiguation words or concepts in one resource by mapping them to the other resource (e.g., mapping Wiktionary entries onto WordNet senses). Thus, the huge body of work on word sense disambiguation (WSD) is relevant, too. Methodologically, this research also relies, to a large extent, on relatedness/similarity measures and random walks on appropriately constructed graphs. See (Navigli, 2009) for an extensive survey.

There is remotely related work on several other tasks in computational linguistics and text mining. These include semantic relatedness between concepts or words (e.g., (Gabrilovich and Markovitch, 2007; Pilehvar et al., 2013)), type inference for the arguments of a phrase (e.g., (Kozareva and Hovy, 2010; Nakashole et al., 2013)), and entailment among verbs (e.g., (Hashimoto et al., 2009)). The SemEval-2010 task on classification of semantic relations (Hendrickx et al., 2010) addressed the problem of predicting the relation for a given sentence and pair of nominals, but was limited to a small prespecified set of relations.

3 Constructing a Candidate Alignment Graph

The general idea of the main algorithm is to align phrase synsets from the Patty taxonomy with verb synsets in WordNet. To this end, we first construct a directed *candidate alignment graph* (CAG). Section 4 will then discuss the actual alignment algorithm.

Vertices of the CAG represent

- synsets of *relational phrases* in Patty, or phrases for short,
- *verb senses* from WordNet, verbs for short,
- *features* of either phrases or verbs.

Edges of the CAG correspond to relations between phrases, verbs, and features. We consider three types of relations here: similarity, hypernymy, and vertex-features. Edges are weighted (see below).

Vertex Types: There are 6 kinds of vertices in the CAG. Since we aim to connect Patty *phrases* with WordNet *verbs*, these two are the main kinds of vertices. Additionally, the graph contains feature vertices representing *noun senses* from WordNet (nouns for short), *surface verbs* as occurring in sample texts, *sentence frames* from WordNet, and specifically derived *phrase-verb vertices* connecting phrases and verbs. The latter are constructed by combining each phrase with its top-10 most similar verb senses. To this end, we retrieve all verb synsets from WordNet and rank the verb synsets by the cosine similarity between the support sentences that Patty provides for its phrases (i.e., sentences from Wikipedia that contain instances of a phrase) and the usage examples in WordNet glosses. The resulting vertices are labeled by the combination of phrase id and verb-sense id. Having these combinations as vertices, rather than simply connecting phrases and verbs via edges, leads to a CAG structure that is better suited for our random walk algorithms (see Section 4). Table 1 gives examples for the 6 vertex types.

Relational Phrase	Verb Sense	Noun Sense	Surface Verb	Sentence Frame	Phrase-Verb Pair
[person] succeeded [person]	succeed2#verb	king1#noun	succeed	Somebody ----s somebody	(phrase_1, verb_sense_2)
[musician] played jazz with [musician]	play3#verb	music1#noun	play	Somebody ----s something	(phrase_2, verb_sense_3)

Table 1: Examples of vertex types

Edge Types: Edges in the graph represent 3 different types of relationships between vertices:

- For all relational phrases, all verb senses from WordNet and also all noun senses (as feature vertices), we capture their *hypernymy relations* as edges.
- We connect phrase-verb vertices with their constituents, phrase vertices and verb vertices, by *similarity edges*, with weights derived from the similarity computation.
- The remaining edges connect phrases or verbs with their respective feature vertices. There are 6 kinds of such *vertex-feature edges*, explained next.

Verb Features: The following features are associated with verb senses. A *lemma edge* connects a verb sense with one or more surface-verb vertices, as given in WordNet glosses. A *domain edge* connects a verb sense with noun senses that describe the usage domain of the verb (e.g. literature, politics). This information is retrieved from WordNet and the WordNet Domains project (Bentivogli et al., 2004). While the latter does not provide sense-disambiguated information, we need to add a mechanism which maps domain information to its WordNet noun sense counterpart. Therefore, we map domain surface nouns to their most frequent senses.

In addition, we harness the WordNet links of type *derivationally related form* to construct further edges between verb senses and noun-sense features in our CAG. The last type of edges for verb-sense features are *sentence frame edges*, between verb vertices and feature vertices of type sentence frame. WordNet for each verb sense provides information about its sentence frames. There are defined 35 possible sentence frames.

Phrase Features: Relational phrases are associated with the following features. A *verb-in-phrase edge* connects a phrase with a surface verb whenever the phrase contains the verb after lemmatization. Analogously to the domain edges for verb senses, we introduce *Wikipedia-category edges* between relational phrases and noun senses. Patty provides us with Wikipedia articles where instances of a phrase occur. We consider all Wikipedia categories of such an article as a source for related noun senses. We use ontological types of the articles and the categories and their mappings to Wordnet provided by the YAGO project (Suchanek et al., 2007). Finally, we also introduce *sentence-frame edges* between relational phrases and sentence-frame feature vertices. To avoid polluting the CAG with overly noisy connections, we apply specific tests. First, we check if the lexical argument types of a phrase and a frame are compatible (e.g., musician is compatible with person, but not with location). Second, we compare characteristic prepositions in the phrase and the frame. We create an edge only if these additional tests are affirmative.

Examples of vertices connected by the different edge types with verb vertices and phrase vertices are shown in Table 2 and 3, respectively.

Hypernymy	Similarity	Lemma	Domain	Derivationally Related Form	Sentence Frame
replace2#verb	(phrase_1, verb_sense_2)	“succeed”, “come after”	politics1#noun	successor1#noun	Somebody ----s somebody

Table 2: Vertices connected by different edges with vertex “succeed2#verb” of type verb.

Hypernymy	Similarity	Verbs in phrase	Wikipedia Category	Sentence Frame
[person] replaced [person]	(phrase_1, verb_sense_2)	“succeed”	politician1#noun	Somebody ----s somebody

Table 3: Vertices connected by different edges with vertex “[person] succeeded [person]” of type phrase.

Edge Weights: All edges in the graph are weighted. The weights are derived from frequency counts of features and/or similarity scores, or are simply set to 1 for binary cases (e.g., hypernymy edges). Lemma edges between verb senses and surface verbs vertices are weighted in proportion to the frequency count of a verb sense, as given by WordNet. Wikipedia-category edges have weights based on the number of occurrences of a relational phrase in Wikipedia articles and the frequencies of categories. Similarity edges have weights set according to the cosine similarity between examples of a verb sense and examples of a relational phrase.

Finally, we normalize all weights in the graph by requiring that the sum of weights of the incoming edges is equal to 1 for every vertex. For the verb and phrase vertices, we perform an additional normalization so that each kind of edge has the same impact in terms of the total edge weight per edge kind.

The above procedure leads to a CAG with 238,437 vertices and 4,776,116 edges. Figure 1 shows an excerpt for illustration.

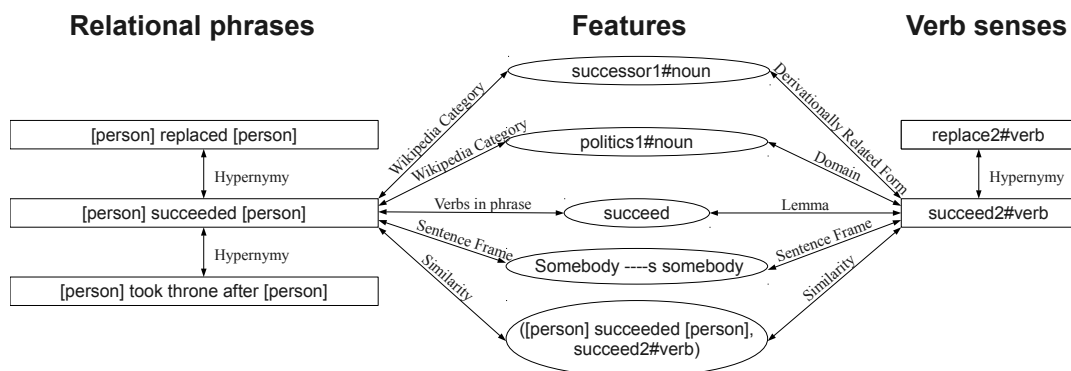


Figure 1: Excerpt from Candidate Alignment Graph

4 Alignment Algorithm

Our algorithm runs on the directed *candidate alignment graph* (CAG). Intuitively, it aims to find “strong paths” between relational-phrase vertices and verb-sense vertices. We use random-walk methods to this end. For each relational phrase, we compute scores and a ranked list of verb senses to which the phrase likely corresponds. The top-ranked verb would ideally be the desired alignment.

SimRank: We employ the SimRank algorithm (Jeh and Widom, 2002), an advanced form of random walks. SimRank computes similarity scores between a pair of vertices in a weighted graph, based on the neighborhoods of the two vertices. The definition, formally given in Equation 1, is recursive: two vertices are similar if their neighborhoods are similar. In the standard SimRank equation, $I_i(a)$ represents the i^{th} (incoming) neighbor of vertex a , and C is a constant dampening factor.

$$s(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \quad (1)$$

SimRank helps capturing long-distance dependencies between vertices in a graph. This would not be achieved by simpler similarity measures of context vectors. Note that SimRank is quite different from (Personalized) PageRank methods; SimRank can be seen as a random walk over pairs of nodes, not over individual nodes. During the CAG construction, we tried to keep the path lengths between phrase vertices and verb vertices uniform for all kinds of feature vertices, to avoid biasing the influence of specific features. Since the SimRank similarity is based on two random walks meeting, the method works best when all paths between source-target node pairs have even length. With this property SimRank produces better results; we introduced explicit phrase-verb vertices for this reason.

SimRank with Fingerprints: Unfortunately, SimRank has very high computational complexity: the run-time of a straightforward implementation is $O(Kn^4)$, where n is the number of vertices in the graph and K is the number of iterations in an iterative fixpoint computation (in the style of the Jacobi method). However, there are much faster approximations of SimRank. We use a variant known as *SimRank with fingerprints* (Fogaras and Racz, 2005). To approximate the SimRank score for two vertices, this method computes the *expected first meeting time* for two random walks originating from the two vertices (with randomized restarts). To this end, the method precomputes a fingerprint for each vertex a : a data structure holding the visiting probabilities of vertices for standard random walks originating in a . A fast implementation actually runs random walks a specified number of times, to estimate the visiting probabilities. For two vertices a and b , the expected number of hops until their random walks meet in a common vertex is then efficiently computed from the fingerprints of a and b . Moreover, this method allows computing the SimRank score for a pair of vertices on demand, only for vertex pairs of interest, rather than having to compute all $O(n^2)$ scores.

The original SimRank method works with unweighted graphs. In our setting, we modify transition probabilities according to edge weights. Our extended SimRank variant is equivalent to Equation 2,

where $W(a, b)$ denotes the weight of the edge between a and b . This equation is similar to the weighted variant of (Antonellis et al., 2007).

$$s_w(a, b) = C * \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} W(a, I_i(a)) * W(b, I_j(b)) * s_w(I_i(a), I_j(b)) \quad (2)$$

Unlike the original SimRank method, we also incorporate random jumps in the underlying random-walk model. Each vertex has a different random jump probability, explained next.

Random Jumps: The original SimRank definition favors vertices with smaller neighborhoods. To avoid this bias, we introduce a form of smoothing on the graph. Whenever a phrase vertex or verb vertex lacks some of the feature types that other vertices may have, we introduce an option for random jumps from the given vertex to any other vertex in the graph. For each missing kind of feature (e.g., domain feature or sentence-frame feature), we assign a probability mass of ϵ , a small constant, for a random jump. So if several features are missing, there is an accumulated probability for a jump. The target of a random jump is always chosen with uniform distribution. A final normalization of edge weights (with linear adjustment) ensures that the possible transitions from a vertex form a proper probability distribution. The method works also without smoothing (i.e., setting the constant to 0), but the results tend to be worse. The results are not very sensitive to the exact choice of the random-jump parameter.

Filtering and Candidate Pruning: The target of our alignment is the WordNet verb hierarchy, but not all relational phrases can be mapped into this target space. Therefore, we restrict ourselves to a subset of relational phrases that contain exactly one verb. This eliminates noun phrases (e.g. “father of”) and phrases that contain multiple verbs (e.g. “succeed and died”, “succeeded in persuading”). Noun phrases should be aligned to the WordNet noun hierarchy and it should be treated as a different task (using e.g. state-of-the-art work (Ponzetto and Navigli, 2010)). Multi-verb phrases often pose semantic difficulties. Note that the verbs in these phrases are always transitive verbs, as *Patty* is derived from subject-phrase-object structures in large corpora. We also used the cardinalities of the support sentences in *Patty* for pruning the noisy tail of phrases, by dropping all phrases that have only a single instance.

To avoid computing SimRank scores for every pair of vertices, we prune the search space as follows. We consider only pairs of relational phrases and verb senses which contain the same surface verb (with lemmatization).

Deriving Hypernymy Links: Once we have alignments between phrases and verbs, we derive hypernymy relations *among phrases* as follows. Whenever phrases p_1 and p_2 are aligned with verb senses v_1 and v_2 , respectively, and v_1 is a direct or transitive hypernym of v_2 , we infer that p_1 is a hypernym of p_2 . We consider transitive hypernyms because not every WordNet verb sense has a phrase aligned with it; without transitivity we would obtain a very sparse hierarchy. By the acyclicity of the WordNet hypernymy structure, the process yields a proper DAG. However, the output contains redundant links (direct ones and transitive ones connecting the same pair of phrases); these are subsequently eliminated by a transitive reduction algorithm (Aho et al., 1972).

5 Evaluation

We evaluated the quality of the HARPY alignments by manual assessment of a large sample set, and compared it against several alternative methods.

Baselines: We compared our SimRank-based method against the following baselines, each given the same feature set:

- *Cosine Similarity:* for each relational phrase and verb sense, we create a contextual vector (in the spirit of distributional semantics) consisting of the features described in Section 3, with tf-idf-based weights (Manning et al., 2008). The alignment ranking is computed by the cosine similarity of tf-idf-weighted contextual vectors.
- *Modified Adsorption (MAD):* a label propagation algorithm (Talukdar and Crammer, 2009) run on the candidate alignment graph. In our setting, each relational phrase is a label. Initially, only the respective phrase vertices have this label. The algorithm propagates labels to other vertices, based on

the graph’s edge weights. The top-k results for the alignment of a phrase are the verb senses with the highest probability for the phrase label. We use the Junto Label Propagation Toolkit ¹.

- *Personalized PageRank (PPR)*: a method for random walks with random jumps back to the start vertex (Haveliwala, 2002). For each phrase, a separate PPR is performed. The ranking of verb senses is produced by the visiting probabilities according to the PPR scores.
- *Most Frequent Sense (MSF)*: For each phrase, we consider only verb senses that contain the same surface verb (with lemmatization), and rank them by the WordNet frequency information.

Assessment: We retrieved a random subset of 261 relational phrases considered for alignment, and showed the results of the different alignment methods to two human judges. For each relational phrase, we displayed its textual form, list of usage examples, and the top-5 ranked list of verb senses computed by each method under comparison. Each verb sense was enriched with information about its lemmas, its gloss, and examples. The evaluators were asked to identify the verb sense that is semantically equivalent to the given relational phrase (including the option of saying “none”).

Quality Measures: As all methods compute a ranked list of verb senses for a given phrase where exactly one list item is correct, we use quality measures geared for such rankings: Mean Reciprocal Rank (*MRR*) and Normalized Discounted Cumulative Gain (*NDCG*). In addition, we report on the precision for top-k results, for small k (1, 3, or 5). Here, a top-k result is considered good if the correct verb senses appears among the top-k alignments, for a given phrase.

Results: The results are shown in Table 4. Our method outperforms all baselines. Among the competitors, MFS shows the best performance. This is not so surprising; MFS is rarely outperformed in word sense disambiguation (McCarthy et al., 2004; Navigli and Lapata, 2010). Our gains over MFS are remarkable. In total, HARPY aligned 20,812 phrases to 4,789 verb senses, and also obtained 616,792 hypernymy links between phrases.

The evaluation process led to high inter-judge agreement, with Cohen’s Kappa around 0.678. The number of samples, 261, was large enough for statistical significance: we performed a paired t-test for *MRR*, *NDCG* and *Precision@1* of the SimRank results against each of the baselines, and obtained p-values below 0.05.

	SimRank	MFS	PPR	MAD	Cosine
MRR	0.698	0.664	0.553	0.463	0.252
NDCG	0.733	0.705	0.584	0.51	0.279
Precision@1	0.571	0.517	0.41	0.318	0.161
Precision@3	0.793	0.778	0.644	0.594	0.307
Precision@5	0.874	0.866	0.736	0.67	0.391

Table 4: Evaluation

Tables 5 and 6 shows example results that HARPY computed. Table 5 has correct outputs. We see that HARPY manages to distinguish between the sport, musical, and theatrical senses of the verb “play”. As shown in Table 6, HARPY also produces some spurious results, with various factors contributing to these errors. For example, the phrase “covered on album” was aligned with the first sense of “cover” since there is no musical sense for “cover” in WordNet. Other errors arise from mistakes in the original Patty repository of relational phrases. For example, the travel sense of the verb “head” was aligned with the phrase “head of” because “head of” and “head to” were in the same Patty synset. Yet another cause of problems is the extremely fine granularity of WordNet: even for humans it is often hard to distinguish between love as a state of liking and love as being enamored.

6 Extrinsic Study: Lexical Types for WordNet Verbs

As an extrinsic use-case for the HARPY resource, we studied the task of inferring lexical types for the subject and object arguments of a WordNet verb sense. For a given verb sense, we propagate the type signature of the relational phrase with the highest alignment score.

¹<http://code.google.com/p/junto/>

Relational phrase	Verb Sense	WordNet definition
[musician] played with [musician]	play3	play on an instrument
[actor] played [[det]] role in [event]	act3	play a role or part
[person] played hockey for [organization]	play1	participate in games or sport
[person] was shooting [person]	shoot2	kill by firing a missile
[movie] be shot in [city]	film1	make a film or photograph of something
[composition] written by [composer]	compose2	write music
[writer] writing at [organization]	write1	produce a literary work

Table 5: Correct examples

Relational phrase	Verb Sense	WordNet definition
[person] covered on album [artifact]	cover1	provide with a covering or cause to be covered
[person] head of [artifact]	head1	to go or travel towards
[person] becomes convinced that [person]	become1	enter or assume a certain state or condition
[person] is loved by [person]	love1	have a great affection or liking for
[wrestler] wrestled in [organization]	wrestle1	combat to overcome an opposing tendency or force

Table 6: Wrong alignment examples

For comparison, this procedure is performed with the HARPY alignments as well as the alignments by the baseline methods. We showed a uniformly sampled set of 261 results to human judges, who assessed as valid or invalid. Additionally, we had a set of the 100 most-confident results (those derived from the highest alignment scores) assessed in the same manner.

For the uniform samples, the type signature derived from HARPY had a precision of 0.46, whereas the best of the baselines (PPR and Cosine) achieved 0.39. For the top-100 samples, HARPY achieved a precision of 0.81. Table 7 shows some example results, demonstrating the added value beyond WordNet.

Domain	Range	Verb Sense	WordNet definition
country	country	export1	sell or transfer abroad
person	country	head2	be in charge of
organization	organization	own1	have ownership or possession of
person	person	predate1	be earlier in time; go back further
saint	organization	reverence1	regard with feelings of respect and reverence
person	artifact	rush5	run with the ball, in football
organization	person	sustain4	supply with necessities and support
musician	musician	play3	play on an instrument
football_player	athlete	pass20	throw (a ball) to another player
singer	composer	inspire2	supply the inspiration for
ruler	country	suppress1	to put down by force or authority
architect	city	design2	plan something for a specific role or purpose or effect
priest	saint	canonize2	treat as a sacred person
country	country	ally_with1	unite formally; of interest groups or countries
company	organization	deal13	sell
artifact	computer_game	port8	modify (software) for use on a different machine or platform

Table 7: Type inference examples by HARPY

7 Conclusion

HARPY is a new resource that aligns lexically typed multi-word phrases for binary relations with WordNet verb senses. By judiciously devising appropriate features and adapting and extending an advanced random-walk method, SimRank, we achieved high-quality alignments, as shown in our evaluation. This creates added value for both the resource of relational phrases, Patty, and WordNet. Phrases are now organized into a clean hypernymy hierarchy, an important aspect on which the Patty work fell short. WordNet verb senses, on the other hand, are extended by a rich set of paraphrases and also by lexical type signatures inherited from the phrases. We believe that this new resource is a useful asset for computational linguistics. As a future work, we plan to align additional resources like WiseNet (Moro and Navigli, 2012), FrameNet (Baker et al., 1998) or VerbNet (Kipper et al., 2008). The HARPY resource is publicly available at www.mpi-inf.mpg.de/yago-naga/patty/.

References

- Alfred V. Aho, M. R. Garey, Jeffrey D. Ullman. 1972. The Transitive Reduction of a Directed Graph. *SIAM J. Comput.*, 131–137.
- Enrique Alfonseca, Daniele Pighin, and Guillermo Garrido. 2013. HEADY: News headline abstraction through event pattern clustering. *ACL (1)*, 1243–1253.
- Ioannis Antonellis, Hector Garcia-Molina, and Chi-Chao Chang. 2007. Simrank++: Query rewriting through link analysis of the click graph. *CoRR*, abs/0712.0499.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. *COLING-ACL*, 86–90.
- Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. *IJCAI*, 805–810.
- Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004. Revising the WordNet Domains hierarchy: Semantics, coverage and balancing. In *Proceedings of the Workshop on Multilingual Linguistic Resources*, MLR '04, 101–108, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1): 13–47.
- Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for Never-Ending Language Learning. *AAAI*
- Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. *EMNLP*, 33–40.
- Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. *CIKM*, 513–522.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. *EMNLP*, 1535–1545.
- Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. *ACL (1)*, 1608–1618.
- Christiane Fellbaum, George Miller (Editors). 1998. *WordNet An Electronic Lexical Database*. The MIT Press.
- Óscar Ferrández, Michael Ellsworth, Rafael Muñoz, and Collin F. Baker. 2010. Aligning FrameNet and WordNet based on semantic neighborhoods. *LREC*.
- Dániel Fogaras and Balázs Rácz. 2005. Scaling link-based similarity search. *WWW*, 641–650.
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *IJCAI*, 1606–1611.
- Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, François Scharffe, Pavel Shvaiko, Cássia Trojahn dos Santos, and Ondrej Zamazal. 2013. Results of the ontology alignment evaluation initiative 2013. *Ontology Matching*, volume 1111 of *CEUR Workshop Proceedings*, 61–100.
- Rahul Gupta, Alon Halevy, Xuezhong Wang, Steven Whang, and Fei Wu. 2014. Biperpedia: An ontology for search applications. *Proc. 40th Int'l Conf. on Very Large Data Bases (PVLDB)*. 505–516 .
- Chikara Hashimoto, Kentaro Torisawa, Kow Kuroda, Stijn De Saeger, Masaki Murata, and Jun'ichi Kazama. 2009. Large-scale verb entailment acquisition from the web. *EMNLP*, 1172–1181.
- Taher H. Haveliwala. 2002. Topic-sensitive PageRank. *WWW*, 517–526.
- Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *Proceedings of SemEval-2*, Uppsala, Sweden.
- Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. *KDD*, 538–543.

- Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40.
- Zornitsa Kozareva and Eduard H. Hovy. 2010. Learning arguments and supertypes of semantic relations using recursive patterns. *ACL*, 1482–1491.
- Dekang Lin and Patrick Pantel. 2001. DIRT @SBT@discovery of inference rules from text. *KDD*, 323–328.
- Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. 2008. Scoring, Term Weighting, and the Vector Space Model. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England, 2008, pp. 109–133.
- Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. *EMNLP-CoNLL*, 523–534.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John A. Carroll. 2004. Finding predominant word senses in untagged text. *ACL*, 279–286.
- Christian M. Meyer and Iryna Gurevych. 2012. To exhibit is not to loiter: A multilingual, sense-disambiguated Wiktionary for measuring verb similarity. *COLING*, 1763–1780.
- Andrea Moro and Roberto Navigli. 2012. WiseNet: building a Wikipedia-based semantic network with ontologized relations. *CIKM*, 1672–1676.
- Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. *EMNLP-CoNLL*, 1135–1145.
- Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. *ACL (1)*, 1488–1497.
- Roberto Navigli and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):678–692.
- Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).
- Martha Palmer. 2009. SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference GenLex-09*, Pisa, Italy, Sept.
- Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. *ACL (1)*, 1341–1351.
- Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-Rich Word Sense Disambiguation Rivaling Supervised Systems. *ACL*, 1522–1531.
- Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. *LREC*, pages 3679–3686.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: a core of semantic knowledge. *WWW*, 697–706.
- Partha Pratim Talukdar and Koby Crammer. 2009. New regularized algorithms for transductive learning. *ECML/PKDD (2)*, volume 5782 of *Lecture Notes in Computer Science*, pages 442–457. Springer.
- Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. *SIGMOD Conference*, 481–492.