Automatic Food Categorization from Large Unlabeled Corpora and Its Impact on Relation Extraction

Michael Wiegand and **Benjamin Roth** and **Dietrich Klakow**
Spoken Language Systems
Saarland University
D-66123 Saarbrücken, Germany
{Michael.Wiegand|Benjamin.Roth|Dietrich.Klakow}@lsv.uni-saarland.de

Abstract

We present a weakly-supervised induction method to assign semantic information to food items. We consider two tasks of categorizations being food-type classification and the distinction of whether a food item is composite or not. The categorizations are induced by a graph-based algorithm applied on a large unlabeled domain-specific corpus. We show that the usage of a domain-specific corpus is vital. We do not only outperform a manually designed open-domain ontology but also prove the usefulness of these categorizations in relation extraction, outperforming state-of-the-art features that include syntactic information and Brown clustering.

1 Introduction

In view of the large interest in food in many parts of the population and the ever increasing amount of new dishes/food items, there is a need of automatic knowledge acquisition. We approach this task with the help of natural language processing.

We investigate different methods to assign categories to food items. We focus on two categorizations, being a classification of food items to categories of the *Food Guide Pyramid* (U.S. Department of Agriculture, 1992) and a categorization of whether a food item is composite or not.

We present a semi-supervised graph-based approach to induce these food categorizations from an unlabeled domain-specific text corpus crawled from the Web. The method only requires minimal manual guidance for the initialization of the algorithm with seed terms. It depends, however, on an automatically constructed high-quality similarity graph. For that we choose a pattern-based representation that outperforms a distributional-based representation. For initialization, we examine some manually compiled seed words and a very few simple surface patterns to automatically induce such expressions. As a hard baseline, we compare the effectiveness of using a general-purpose ontology for the same types of categorizations. Apart from an intrinsic evaluation, we also examine the categories in relation extraction.

The contributions of this paper are a method requiring minimal supervision for a comprehensive classification of food items and a proof of concept that the knowledge that can thus be gained is beneficial for relation extraction. Even though we focus on a specific domain, the induction method can be easily translated to other domains. In particular, other life-style domains, such as fashion, cosmetics or home & gardening, show parallels since comparable textual web data are available and similar relation types (e.g. that two items fit together or can be substituted by each other) exist.

Our experiments are carried out on German data but our findings should carry over to other languages since the issues we address are (mostly) language universal. For general accessibility, all examples are given as English translations.

2 Data & Annotation

2.1 Domain-Specific Text Corpus

In order to generate a dataset for our experiments, we used a crawl of *chefkoch.de*[1] (Wiegand et al., 2012b) consisting of $418,558$ webpages of food-related forum entries. *chefkoch.de* is the largest German web portal for food-related issues.

2.2 Food Categorization

As a food vocabulary, we employ a list of 1888 food items: 1104 items were directly extracted from GermaNet (Hamp and Feldweg, 1997), the German version of WordNet (Miller et al., 1990). The items were identified by extracting all hyponyms of the synset *Nahrung* (English: *food*). By

[1] www.chefkoch.de

Class	Description	Size	Perc.
MEAT	meat and fish (products)	394	20.87
BEVERAGE	beverages (incl. alcoholic drinks)	298	15.78
VEGE	vegetables (incl. salads)	231	12.24
SWEET	sweets, pastries and snack mixes	228	12.08
SPICE	spices and sauces	216	11.44
STARCH	starch-based side dishes	185	9.80
MILK	milk products	104	5.51
FRUIT	fruits	94	4.98
GRAIN	grains, nuts and seeds	77	4.08
FAT	fat	41	2.18
EGG	eggs	20	1.06

Table 1: The different food types (*gold standard*).

consulting the relation tuples from Wiegand et al. (2012c) a further 784 items were added. We manually annotated this vocabulary w.r.t. two tasks:

2.2.1 Task I: Food Types

The food type categories we chose are mainly inspired by the *Food Guide Pyramid* (U.S. Department of Agriculture, 1992) that divides food items into categories with similar nutritional properties. This categorization scheme not only divides the set of food items in many intuitive homogeneous classes but it is also the scheme that is most commonly agreed upon. Table 1 lists the specific categories we use. For category assignment of complex dishes comprising different food items we applied a heuristics: we always assign the category that dominates the dish. A *meat sauce*, for example, would thus be assigned MEAT (even though there may be other ingredients than meat).

2.2.2 Task II: Dishes vs. Atomic Food Items

In addition to Task I, we include another categorization that divides food items into dishes and atomic food items (Table 2). By dish, we mainly understand food items that are composite food items made of other (*atomic*) food items. This categorization is orthogonal to the previous classification of food items. We refrained from adding dishes as a further category of food types in §2.2.1, as we would have ended up with a very heterogeneous class in the set of homogeneous food type categories. Thus, dishes that differ greatly in nutrient content, such as *Waldorf salad* and *chocolate cake*, would have been subsumed by one class.

Class	Description	Examples	Perc.
DISH	composite food items	*cake, falafel, meat loaf*	32.10
ATOM	non-composite food items	*apple, steak, potato*	67.90

Table 2: Distribution of dishes and atomic food items among the food vocabulary (*gold standard*).

3 Method

3.1 Graph-based Induction

We propose a semi-supervised graph-based approach to label food items with their respective food categories. The underlying data structure is a similarity graph connecting different food items. Food items that belong to the same category should be connected by highly weighted edges. In order to infer the labels for each respective food item, one first needs to specify a small set of seeds for each category and then apply a graph-based clustering method that divides the graph into clusters that represent distinct food categories. Our method is a low-resource approach that can also be easily adapted to other domains. The only domain-specific information required are an unlabeled corpus and a set of seeds.

3.1.1 Construction of the Similarity Graph

To enable a graph-based induction, we generate a similarity graph that connects similar food items. For that purpose, a list of *domain-independent* similarity-patterns was compiled. Each pattern is a lexical sequence that connects the mention of two food items (Table 3). Each pair of food items observed with any of those patterns is connected via a weighted edge (the different patterns are treated equally). The weight is the total frequency of all patterns co-occurring with a particular food pair.

Due to the high precision of our patterns, with one or a few prototypical seeds we cannot expect to find all items of a food category within the set of items to which the seeds are *directly* connected. Instead, one also needs to consider transitive connectedness within the graph. For example, in Figure 1 *banana* and *redberry* are not directly connected but they can be reached via *pear* or *raspberry*. However, by considering mediate relationships it becomes more difficult to determine the most appropriate category for each food item since most food items are connected to food items of different categories (in Figure 1, there are not only edges between *banana* and other types of fruits but there is also some edge to some sweet, i.e. *chocolate*). For a unique class assignment, we apply a robust graph-based clustering algorithm. (It will figure out that *banana*, *pear*, *raspberry* and *redberry* belong to the same category and *chocolate* belongs to another category, since it is mostly

Patterns	food_item$_1$ (*or*|*or rather*|*instead of*|"(") food_item$_2$
Example	{*apple*: *pineapple, pear, fruit, strawberry, kiwi*} {*steak*: *schnitzel, sausage, roast, meat loaf, cutlet*}

Table 3: *Domain-independent* patterns for building the similarity graph.

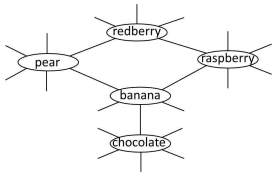

Figure 1: Illustration of the similarity graph.

linked to many other food items not being fruits.)

3.1.2 Semi-Supervised Graph Optimization

Our semi-supervised graph optimization (Belkin and Niyogi, 2004) is a robust algorithm that was primarily chosen since it only contains few free parameters to adjust. It is based on two principles: First, similar data points should be assigned similar labels, as expressed by a similarity graph of labeled and unlabeled data. Second, for labeled data points the prediction of the learnt classifier should be consistent with the (actual) gold labels.

We construct a weighted transition matrix W of the graph by normalization of the matrix with co-occurrence counts C which we obtain from the similarity graph (§3.1.1). We use the common normalization by a power of the degree function $d_i = \sum_j C_{ij}$: it defines $W_{ij} = \frac{C_{ij}}{d_i^\lambda d_j^\lambda}$ if $i \neq j$, and $W_{ii} = 0$. The normalization weight λ is the first of two parameters used in our experiments for semi-supervised graph optimization. For learning the semi-supervised classifier, we use the method of Zhou et al. (2004) to find a classifying function which is sufficiently smooth with respect to both the structure of unlabeled and labeled points.

Given a set of data points $\mathcal{X} = \{x_1, \ldots, x_n\}$ and label set $\mathcal{L} = \{1, \ldots, c\}$, with $x_{i:1 \leq i \leq l}$ labeled as $y_i \in \mathcal{L}$ and $x_{i:l+1 \leq i \leq n}$ unlabeled. For prediction, a vectorial function $F : \mathcal{X} \rightarrow \mathbb{R}^c$ is estimated assigning a vector F_i of label scores to every x_i. The predicted labeling follows from these scores as $\hat{y}_i = \arg\max_{j \leq c} F_{ij}$. Conversely, the gold labeling matrix Y is a $n \times c$ matrix with $Y_{ij} = 1$ if x_i is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise.

Minimizing the cost function $\mathcal{Q}$ aims at a trade-off between information from neighbours and initial labeling information, controlled by parameter μ (the second parameter used in our experiments):

$$\mathcal{Q} = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{\delta_i}} F_i - \frac{1}{\sqrt{\delta_j}} F_j \right\| + \mu \sum_{i=1}^{n} \|F_i - Y_i\|\right)$$

where δ_i is the degree function of W.

The first term in $\mathcal{Q}$ is the smoothness constraint, its minimization leads to adjacent edges having similar labels. The second term is the fitting constraint, its minimization leads to consistency of the function F with the labeling of the data. The solution to the above cost function is found by solving a system of linear equations (Zhou et al., 2004).

As we do not possess development data for this work, we set the two free parameters $\lambda = 0.5$ and $\mu = 0.01$. This setting is used for both induction tasks and all configurations. It is a setting that provided reasonable results without any notable bias for any particular configuration we examine.

Patterns	Categorization	Examples
patt$_{hearst}$	Food Types	food_item *is some* food_type, food_type *such as* food_item, . . .
patt$_{dishes}$	Dishes	*recipe for* food_item
patt$_{atom}$	Atomic Food Items	*made of*|*contains* food_item

Table 4: List of patterns to extract seeds.

3.1.3 Manually vs. Automatically Extracted Seeds

We explore two types of seed initializations: (a) a manually compiled list of seed food items and (b) a small set of patterns (Table 4) by the help of which such seeds are automatically extracted.

In order to extract seeds for Task I with the pattern-based approach, we apply the patterns from Hearst (1992). These patterns have been designed for the acquisition of hyponyms. Task I can also be regarded as some type of hyponym extraction. The food types (*fruit*, *meat*, *sweets*) represent the hypernyms for which we extract seed hyponyms (*banana*, *beef*, *chocolate*).

In order to extract seeds for Task II, we apply two domain-specific sets of patterns (*patt$_{dish}$* and *patt$_{atom}$*). We rank the food items according to the frequency of occurring with the respective pattern set. Since food items may occur in both rankings, we merge the two rankings in the following way:

$$score(food\,item) = \#patt_{dish}(food\,it.) - \#patt_{atom}(food\,it.)$$

The top end of this ranking represents dishes while the bottom end represents atoms.

3.2 Using a General-Purpose Ontology

As a hard baseline, we also make use of the semantic relationships encoded in GermaNet. Our two

types of food categorization schemes can be approximated with the hypernymy graph in that ontology: We manually identify nodes that resemble our food categories (e.g. *fruit*, *meat* or *dish*) and label any food item that is an immediate or a mediate hyponym of these nodes (e.g. *apple* for *fruit*) with the respective category label. The downside of this method is that a large amount of food items is missing from the GermaNet-database (§2.2).

3.3 Other Baselines & Post-Processing

In addition to the previous methods we implement a heuristic baseline (**HEUR**) that rests on the observation that German food items of the same food category often share the same suffix, e.g. *Schokoladenkuchen* (English: *chocolate cake*) and *Apfelkuchen* (English: *apple pie*). For HEUR, we manually compiled a set of few typical suffixes for each food type/dish category (ranging from 3 to 8 suffixes per category). For classification of a food item, we assign the food item the category label whose suffix matched with the food item.[2]

We also examine an *unsupervised* baseline (**UNSUP**) that applies spectral clustering on the similarity graph following von Luxburg (2007):

- Input: a similarity matrix W and the number of categories to detect k.
- The laplacian L is constructed from W. It is the symmetric laplacian $L = I - D^{1/2}WD^{1/2}$, where D is a diagonal degree matrix.[3]
- A matrix $U \in \mathbb{R}^{n \times k}$ is constructed that contains as columns the first k eigenvectors $u_1, \ldots, u_k$ of L.
- The rows of U are interpreted as the new data points. The final clustering is obtained by k-means clustering of the rows of U.

UNSUP (which is completely parameter-free) gives some indication about the intrinsic expressiveness of the similarity graph as it lacks any guidance towards the categories to be predicted.

In graph-based food categorization, one can only make predictions for food items that are connected (be it directly or indirectly) to seed food items within the similarity graph. To expand labels to unconnected food items, we apply some post-processing (**POSTP**). Similarly to HEUR, it exploits the suffix-similarity of food items. It assigns each unconnected food item the label of the food item (that could be labeled by the graph optimization) that shares the longest suffix. Due to their similar nature, we refrain from applying POSTP on HEUR as it would produce no changes.

[2]Unlike German food items, English food items are often multi-word expressions. Therefore, we assume that for English, instead of analyzing suffixes the usage of the head of a multiword expression (i.e. *chocolate cake*) would be an appropriate basis for a similar heuristic.

[3]That is, D_{ii} equals to the sum of the ith row.

		PLAIN				+POSTP			
Configuration	**graph**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**
UNSUP	✓	46.2	43.1	35.7	36.0	56.1	41.0	42.5	38.4
HEUR (plain)		25.5	87.9	32.2	42.9	*N/A*	*N/A*	*N/A*	*N/A*
HEUR	✓	56.4	73.6	52.1	54.7	68.7	72.3	64.3	60.7
PAT-Top1	✓	52.4	60.2	51.2	52.5	64.5	58.2	62.9	57.4
PAT-Top5	✓	61.1	70.7	61.9	64.4	74.5	67.9	76.0	69.7
PAT-Top10	✓	60.2	69.6	60.5	62.2	73.4	66.7	74.2	67.3
1-PROTO	✓	58.0	68.0	58.0	59.5	70.2	64.1	71.0	63.8
5-PROTO	✓	64.5	76.6	63.7	68.6	78.6	73.8	78.5	75.2
10-PROTO	✓	65.8	79.0	**65.5**	71.0	80.2	75.9	**80.6**	77.7
GermaNet (plain)		52.1	**94.0**	52.0	65.7	75.4	73.2	75.0	72.4
GermaNet	✓	**68.3**	84.7	63.4	**71.6**	**82.7**	**81.8**	77.7	**79.1**

Table 5: Comparison of different food-type classifiers (*graph* indicates graph-based optimization).

4 Experiments

We report precision, recall and F-score and accuracy.[4] For precision, recall and F-score, we list the macro-averaged score.

4.1 Evaluation of Food Categorization

4.1.1 Detection of Food Types

Table 5 compares different classifiers and configurations for the prediction of food types (against the gold standard from Table 1). Apart from the previously described baselines, we consider n manually selected prototypes (***n*-PROTO**) and the top n food items produced by Hearst-patterns (**PAT-Top*n***) as seeds for graph-based optimization. The table shows that the semi-supervised graph-based approach with these seeds outperforms the baselines UNSUP and HEUR. Only as few as 5 prototypical seeds (per category) are required to obtain performance that is even better than using plain GermaNet. The table also shows that post-processing (with our suffix-heuristics) consistently improves performance. Manually choosing prototypes is more effective than instantiating seeds via Hearst-patterns. The quality of the output of Hearst-patterns degrades from top 10 onwards. However, considering that PAT-Top*n* does not include any manual intervention, it already produces decent results. Finally, even GermaNet can be effectively used as seeds.

4.1.2 Detection of Dishes

Table 6 compares different classifiers for the detection of dishes (against the gold standard from Table 2). Dishes and atomic food items are very

[4]All manually labeled resources are available at: `www.lsv.uni-saarland.de/personalPages/michael/relFood.html`

		PLAIN				+POSTP			
Configuration	**graph**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**
UNSUP	✓	54.5	59.6	40.2	37.3	67.9	59.0	50.0	40.6
HEUR (plain)		**74.1**	**84.3**	59.9	58.6	*N/A*	*N/A*	*N/A*	*N/A*
PAT-Top25	✓	59.7	72.2	54.6	61.9	74.1	70.1	67.6	68.4
PAT-Top50	✓	60.9	74.4	55.6	63.1	75.9	72.7	69.2	70.3
PAT-Top100	✓	62.7	77.6	57.2	65.2	78.4	76.5	71.5	73.0
PAT-Top250	✓	59.6	71.8	55.1	62.2	74.2	70.3	68.7	69.3
RAND-25	✓	61.4	77.1	54.3	61.8	76.1	74.4	67.1	68.4
RAND-50	✓	62.6	76.3	60.1	67.2	77.2	74.0	76.8	74.4
RAND-100	✓	66.5	82.7	**63.0**	**71.3**	**83.0**	**80.8**	**79.5**	**80.1**
GermaNet (plain)		49.5	81.3	46.5	59.3	79.0	75.9	75.5	75.7
GermaNet	✓	60.8	79.4	51.3	57.6	75.9	78.2	64.4	65.4

Table 6: Comparison of different classifiers distinguishing between dishes and atomic food items (*graph* indicates graph-based optimization).

		PLAIN				+POSTP			
Configuration	**graph**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**
PAT-Top100 (plain)		9.5	89.5	10.5	18.6	63.6	61.5	63.5	61.3
PAT-Top100	✓	62.7	77.6	57.2	65.2	78.4	76.5	71.5	73.0
RAND-100 (plain)		10.6	**100.0**	12.2	21.4	70.2	69.7	69.0	69.0
RAND-100	✓	**66.5**	82.7	**63.0**	**71.3**	**83.0**	**80.8**	**79.5**	**80.1**

Table 7: Impact of graph-based optimization (*graph*) for the detection of dishes.

heterogeneous classes which is why more seeds are required for initialization. This means that we cannot look for *prototypes*. For simplicity, we resorted to randomly sample seeds from our gold standard (**RAND-*n***). For HEUR, we could not find a small and intuitive set of suffixes that are shared by *many* atomic food types, therefore we considered all food types from our vocabulary whose suffix did not match a typical dish suffix as atomic. As this leaves no unspecified food items in our vocabulary, we cannot use the output of HEUR as seeds for graph-based optimization.

In contrast to the previous experiment, HEUR is a more robust baseline. But again, post-processing mostly improves performance, and patterns are not as good as manual (random) seeds yet the former are notably better than HEUR w.r.t. F-Score. Unlike in the food-type classification, graph-based optimization applied on GermaNet does not result in some improvement. We assume that the precision of plain GermaNet with 81.3% is too low.[5]

Since GermaNet cannot effectively be used as seeds for the graph-based optimization and post-processing has already a strong positive effect, we may wonder how effective the actual graph-based optimization is for this classification task. After all, significantly more seeds are required for this classification task than for the previous task, so we need to show that it is not the mere seeds (+post-processing) that are required for a reasonable categorization. Table 7 examines two key configurations *with* and *without* graph-based optimization. It shows that also for this classification task, graph-based optimization produces a categorization superior to the mere seeds. Moreover, the suffix-based post-processing is complementary to the improvement by the graph-based optimization.

4.1.3 Comparison of Initialization Methods

Table 8 compares for each food type 5 manually selected prototypical seeds (i.e. 5-PROTO) and the 5 food items most frequently been observed with patt$_{hearst}$ (Table 4). While the manually chosen seeds represent the spectrum of food items within each particular class (e.g. for STARCH, some type of pasta, rice and potato was chosen), it is not possible to enforce such diversity with the automatically extracted seeds. However, most food items are correct. Table 9 displays the 10 most highly ranked dishes and atomic food items extracted with patt$_{dish}$ and patt$_{atom}$ (Table 4). Unlike the previous task (Table 8), we obtain more heterogeneous seeds within the same class.

4.1.4 Distributional Similarity

Since many recent methods for related tasks, such as noun classification, are based on so-called *distributional similarity* (Riloff and Shepherd, 1997; Lin, 1998; Snow et al., 2004; Weeds et al., 2004; Yamada et al., 2009; Huang and Riloff, 2010; Lenci and Benotto, 2012), we also examine this as an alternative representation to the pattern-based similarity graph (Table 3). We represent each food item as a vector which itself is an aggregate of the contexts of all mentions of a particular food item. We weighted the individual (context) words co-occurring with the food item at a fixed window size of 5 words with *tf-idf*. We can now apply graph-based optimization on the similarity matrix encoding the cosine similarities between any possible pair of vectors representing two food items. As seeds, we use the best configuration (not employing GermaNet), i.e. *10-PROTO* for food type classification and *RAND-100* for the dish classification. Since, however, the graph clustering is not actually necessary, as we have a full similarity matrix (rather than a sparse graph) that also al-

[5]For other seeds for which it worked, we usually measured a precision of 90% or higher.

Class	5 Manually Chosen Seeds (5-PROTO)	5 Hearst-Pattern Seeds (PAT-Top5)
MEAT	*schnitzel, rissole, bologna, redfish, trout*	*salmon, beef, chicken, turkey hen, poultry*
BEVERAGE	*coffee, tea, water, beer, coke*	*coffee, beer, mineral water, lemonade, tea*
VEGE	*peas, green salad, tomato, cauliflower, carrot*	*zucchini, lamb's salad, broccoli, leek, cauliflower*
SWEET	*chocolate, torte, popcorn, apple pie, potato crisps*	*wine gum, marzipan, custard, pancake, biscuits*
SPICE	*pepper, cinnamon, salt, gravy, remoulade*	*cinnamon, laurel, clove, tomato sauce, basil*
STARCH	*spaghetti, basmati rice, white bread, potato, french fries*	*au gratin potatoes, jacket potato, potato, pita, jam*
MILK	*yoghurt, gouda, cream cheese, cream, butter milk*	*butter milk, bovine milk, soured milk, goat cheese, sour cream*
FRUIT	*banana, apple, strawberries, apricot, orange*	*banana, strawberries, pear, melon, kiwi*
GRAIN	*hazelnut, pumpkin seed, rye flour, semolina, wheat*	*sesame, spelt, wheat, millet, barley*
FAT	*margarine, lard, colza oil, spread, butter*	*margarine, lard, resolidified butter, coconut oil, tartar*
EGG	*scrambled eggs, fried eggs, chicken egg, omelette, pickled egg*	*yolk, fried eggs, albumen, offal, easter egg*

Table 8: Comparison of different seed initializations for the food type categorization task (underlined food items represent erroneously extracted food items).

lows us to compare any arbitrary pair of food items *directly*, we also employ a second classifier (for comparison) based on the *nearest neighbour* principle. We assign each food item the label of the most similar seed food item.

Table 10 compares these two classifiers with the best previous result. It shows that the pattern-based representation consistently outperforms the distributional representation. The former may be sparse but it produces high-precision similarity links.[6] The vector representation, on the other hand, may not be sparse but it contains a high degree of noise. The major problem is that not only vectors of similar food items, such as *chips* (*fries*), *potatoes* and *rice*, are similar to each other, but also vectors of different food items that are typically consumed with each other (e.g. *fish* and *chips*). This is because of their frequent co-occurrence (as in collocations like *fish & chips*). Unfortunately, these pairs belong to different food types. For the dish classification, however, the vector representation is less of a problem.[7]

The distributional representation works better with the simple nearest neighbour classifier. We assume that graph-based optimization adds further noise to the classification since, unlike the nearest neighbour which only calculates the *direct* similarity between two vectors, it also incorporates indirect relationships (which may be more error-prone than the direct relationships) between food items.

4.1.5 Do we need a domain-specific corpus?

In this section, we want to provide evidence that apart from the similarity graph and seeds the textual source for the graph, i.e. our domain-specific corpus (*chefkoch.de*), is also important. For that purpose, we compare our current corpus against an open-domain corpus. We consider the German version of *Wikipedia* since this resource also contains encyclopedic knowledge about food items. Table 11 compares the graph-based induction. As in the previous section, we only consider the best previous configuration. The table clearly shows that our domain-specific text corpus is a more effective resource for our purpose than *Wikipedia*.

[6] By the label propagation within the graph-based optimization, the sparsity problem is also mitigated.

[7] *Fish* and *chips* are both atoms, so in the dish classification, it is no mistake to consider them similar food items.

Class	10 Seeds Extracted with Patterns (PAT-Top10)
DISH	*cookies, cake, praline, bread dumpling, jam, biscuit, cheese cake, black-and-whites, onion tart, pasta salad*
ATOM	*marzipan, flour, potato, olive oil, water, sugar, cream, chocolate, milk, tomato*

Table 9: Illustration of seed initialization for the distinction between dishes and atomic food items.

Task	Similarity	Classifier	Acc	F1
Food Type	distributional	nearest neighbour	53.4	51.1
	distributional	graph	25.6	25.6
	pattern-based	graph	**80.2**	**77.7**
Dish	distributional	nearest neighbour	76.8	75.2
	distributional	graph	71.5	71.2
	pattern-based	graph	**83.0**	**80.1**

Table 10: Impact of the similarity representation.

4.2 Evaluation for Relation Extraction

We now examine whether automatic food categorization can be harnessed for relation extraction. The task is to detect instances of the relation types *SuitsTo*, *SubstitutedBy* and *IngredientOf* introduced Wiegand et al. (2012b) (repeated in Table 12) and motivated in Wiegand et al. (2012a). These relation types are highly relevant for customer advice/product recommendation. In particular, *SuitsTo* and *SubstitutedBy* are fairly domain-independent relation types. Customers want to

know which items can be used together (*SuitsTo*), be it two food items that can be used as a meal or two fashion items that can be worn together. Substitutes are also relevant for situations in which item A is out of stock but item B can be offered as an alternative. Therefore, insights from this work should carry over to other domains.

We randomly extracted 1500 sentences from our text corpus (§2.1) in which (at least) two food items co-occur. Each food pair mention was manually assigned one label. In addition to the three relation types from above, we introduce the label *Other* for cases in which either another relation between the target food items is expressed or the co-occurrence is co-incidental. On a subset of 200 sentences, we measured a *substantial* inter-annotation agreement of Cohen's $\kappa = 0.67$ (Landis and Koch, 1977).

We train a supervised classifier and incorporate the knowledge induced from our domain-specific corpus as features. We chose Support Vector Machines with 5-fold cross-validation using *SVM^{light}-multi-class* (Joachims, 1999).

Table 13 displays all features that we examine for supervised classification. Most features are widely used throughout different NLP tasks. One special feature *brown* takes into consideration the output of *Brown clustering* (Brown et al., 1992) which like our graph-based optimization produces a corpus-driven categorization of words. Similar to *UNSUP*, this method is unsupervised but it considers the entire vocabulary of our text corpus rather than only food items. Therefore, this information can be considered as a generalization of all contextual words. Such type of information has been shown to be useful for named-entity recognition (Turian et al., 2010) and relation extraction (Plank and Moschitti, 2013).

For syntactic parsing, Stanford Parser (Rafferty and Manning, 2008) was used. For Brown clustering, the SRILM-toolkit (Stolcke, 2002) was used. Following Turian et al. (2010), we induced 1000 clusters (from our domain-specific corpus §2.1).

4.2.1 Why should food categories be helpful for relation extraction?

All relation types we consider comprise pairs of two food items which makes these relation types likely to be confused. Contextual information may be used for disambiguation but there may also be frequent contexts that are not sufficiently informative. For example, 25% of the instances of *IngredientOf* follow the lexical pattern *food_item$_1$ with food_item$_2$* (1). However, the same pattern also covers 15% of the instances of *SuitsTo* (2).

			PLAIN		+POSTP	
Task	**Corpus**	**graph**	**Acc**	**F1**	**Acc**	**F1**
Food Type	Wikipedia	✓	40.3	49.4	61.4	59.8
	chefkoch.de	✓	65.8	**71.0**	80.2	**77.7**
Dish	Wikipedia	✓	50.4	53.1	75.4	71.1
	chefkoch.de	✓	66.5	**71.3**	83.0	**80.1**

Table 11: Comparison of Wikipedia and domain-specific corpus as a source for the similarity graph.

(1) We had a stew with red lentils. *(Relation: IngredientOf)*
(2) We had salmon with broccoli. *(Relation: SuitsTo)*

The food type information we learned from our text corpus might tell us which of the food items are dishes. Only in (1), there is a dish, i.e. *stew*. So, one may infer that the presence of dishes is indicative of *IngredientOf* rather than *SuitsTo*.

food_item$_1$ and food_item$_2$ is another ambiguous context. It cannot only be observed with the relation *SuitsTo*, as in (3) (66% of all instantiations of that pattern), but also *SubstitutedBy* (20% of all mentions of that relation match that pattern), as in (4). For *SuitsTo*, two food items that belong to two different classes (e.g. *MEAT* and *STARCH* or *MEAT* and *VEGE*) are quite characteristic. For *SubstitutedBy*, the two food items are very often of the same category of the *Food Guide Pyramid*.

(3) I very often eat fish and chips. *(Relation: SuitsTo)*
(4) For these types of dishes you can offer both Burgundy wine and Champagne. *(Relation: SubstitutedBy)*

Since the second ambiguous context involves the two general relation types *SuitsTo* and *SubstitutedBy*, resolving this ambiguity with automatically induced type information has some significance for other domains. In particular, for other life-style domains, domain-specific type information could be obtained following our method from §3.1. The disambiguation rule that two entities of the same type imply *SubstitutedBy* otherwise they imply *SuitsTo* should also be widely applicable.

4.2.2 Results

Table 14 displays the performance of the different feature sets for relation extraction. The features designed from graph-based induction (i.e. *graph*) work slightly better than GermaNet. The performance of *patt* is not impressively high. However, one should consider that *patt* can be used directly without a supervised classifier (as each pattern is

Relation	Description	Example	Freq.	Perc.
SuitsTo	food items that are typically consumed together	My kids love the simple combination of fish fingers with mashed potatoes.	633	42.20
SubstitutedBy	similar food items commonly consumed in the same situations	We usually buy margarine instead of butter.	336	22.40
IngredientOf	ingredient of a particular dish	Falafel is made of chickpeas.	246	16.40
Other	other relation *or* co-occurrence of food items are co-incidental	On my shopping list, I've got bread, cauliflower, ...	285	19.00

Table 12: The different relation types and their respective frequency on our dataset.

Features	Description
patt	lexical surface patterns used in Wiegand et al. (2012b)
word	bag-of-words features: all words within the sentence
brown	features using Brown clustering: all features from *word* but words are replaced by induced clusters
pos	part-of-speech sequence between target food items and tags of the words immediately preceding and following them
synt	path from syntactic parse tree from first target food item to second target food item
conj	conjunctive features: *patt* with brown classes of target food items; *pos* sequence with brown classes of target food items; *synt* with brown classes of target food items
graph	semantic food information induced by graph optimization (config.: *10-PROTO(+POSTP)* and *RAND-100(+POSTP)*)
germanet	semantic food information derived from (plain) GermaNet

Table 13: Description of the feature set.

Features	Acc	Prec	Rec	F1
germanet	45.3	41.3	37.2	37.3
graph	46.0	39.4	39.7	38.6
patt	59.8	49.8	41.1	38.7
word	60.1	56.9	54.5	55.1
word+patt	60.3	57.3	54.9	55.5
word+brown	59.5	56.1	54.6	54.9
word+synt	60.3	57.7	55.4	56.0
word+pos	59.8	56.6	54.6	55.1
word+germanet	61.3	58.6	56.0	56.7
word+graph	62.9	59.2	57.6	58.1°
word+patt+brown+synt+pos	60.4	57.3	56.2	56.5
word+patt+brown+synt+pos+conj	61.7	59.0	57.8	58.2*
word+patt+brown+synt+pos+conj+germanet	63.1	60.2	58.6	59.1°
word+patt+brown+synt+pos+conj+graph	**64.7**	**62.1**	**60.3**	**60.9**$^{\circ\dagger}$

statistical significance testing (paired t-test): better than *word* * at $p < 0.1$/ $^{\circ}$ at $p < 0.05$; † better than *word+patt+brown+synt+pos+conj* at $p < 0.05$

Table 14: Comparison of various features (Table 13) for (unrestricted) relation extraction.

designed for a particular relation type, one can read off from the matching pattern which class is predicted). *word* is slightly better but, unlike *patt*, it is dependent on supervised learning.

The only feature that individually manages to significantly outperform *word* is *graph*. The traditional features (i.e. *pos*, *synt* and *brown*) only produce some mild improvement when added jointly to *word* along some conjunctive features. When *graph* is added to this feature set (i.e. *word+patt+pos+synt+brown+conj*), we obtain another significant improvement. In conclusion, the information we induced from our domain-specific corpus cannot be obtained by other NLP-features, including other state-of-the-art induction methods such as Brown clustering.

5 Related Work

While many of the previous works on noun categorization also address the task of hypernym classification (Hearst, 1992; Caraballo, 1999; Widdows, 2003; Kozareva et al., 2008; Huang and Riloff, 2010; Lenci and Benotto, 2012) and some include examples involving food items (Widdows and Dorow, 2002; Cederberg and Widdows, 2003), only van Hage et al. (2005) and van Hage et al. (2006) specifically focus on the classification of food items. van Hage et al. (2005) deal with ontology mapping whereas van Hage et al. (2006) explore part-whole relations.

The task of data-driven lexicon expansion has also been explored before (Kanayama and Nasukawa, 2006; Das and Smith, 2012), however, our paper presents the first attempt to carry out a *comprehensive* categorization for the food domain. For the first time, we also show that type information can effectively improve the extraction of very common relations. For the twitter domain, the usage of type information based on clustering has already been found effective for supervised learning (Bergsma et al., 2013).

6 Conclusion

We presented an induction method to assign semantic information to food items. We considered two types of categorizations being food-type information and information about whether a food item is composite or not. The categorization is induced by graph-based optimization applied on a large unlabeled domain-specific text corpus. We produce categorizations that outperform a manually compiled resource. The usage of such a domain-specific corpus based on a pattern-based representation is vital and largely outperforms other text corpora or a distributional representation. The induced knowledge improves relation extraction.

Acknowledgements

This work was performed in the context of the Software-Cluster project SINNODIUM. Michael Wiegand was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IC12SO1X. Benjamin Roth is a recipient of the Google Europe Fellowship in Natural Language Processing, and this research is supported in part by this Google Fellowship. The authors would like to thank Stephanie Köser for annotating the dataset presented in this paper.

References

Mikhail Belkin and Partha Niyogi. 2004. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56(1-3):209–239.

Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. 2013. Broadly Improving User Classification via Communication-Based Name and Location Clustering on Twitter. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL)*, pages 1010–1019, Atlanta, GA, USA.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 120–126, College Park, MD, USA.

Scott Cederberg and Dominic Widdows. 2003. Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 111–118, Edmonton, Alberta, Canada.

Dipanjan Das and Noah A. Smith. 2012. Graph-Based Lexicon Expansion with Sparsity-Inducing Penalties. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL)*, pages 677–687, Montréal, Quebec, Canada.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 539–545, Nantes, France.

Ruihong Huang and Ellen Riloff. 2010. Inducing Domain-specific Semantic Class Taggers from (almost) Nothing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 275–285, Uppsala, Sweden.

Thorsten Joachims. 1999. Making Large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press.

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–363, Sydney, Australia.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1048–1056, Columbus, OH, USA.

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

Alessandro Lenci and Guilia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 75–79, Montréal, Quebec, Canada.

Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics (ACL/COLING)*, pages 768–774, Montreal, Quebec, Canada.

George Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3:235–244.

Barbara Plank and Alessandro Moschitti. 2013. Embedding Semantic Similarity in Tree Kernels for Domain Adapation of Relation Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1498–1507, Sofia, Bulgaria.

Anna Rafferty and Christopher D. Manning. 2008. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the ACL Workshop on Parsing German (PaGe)*, pages 40–46, Columbus, OH, USA.

Ellen Riloff and Jessica Shepherd. 1997. A Corpus-Based Approach for Building Semantic Lexicons. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 117–124, Providence, RI, USA.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, British Columbia, Canada.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, Denver, CO, USA.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 384–394, Uppsala, Sweden.

Human Nutrition Information Service U.S. Department of Agriculture. 1992. The Food Guide Pyramid. Home and Garden Bulletin 252, Washington, D.C., USA.

Willem Robert van Hage, Sophia Katrenko, and Guus Schreiber. 2005. A Method to Combine Linguistic Ontology-Mapping Techniques. In *Proceedings of International Semantic Web Conference (ISWC)*, pages 732 – 744, Galway, Ireland. Springer.

Willem Robert van Hage, Hap Kolb, and Guus Schreiber. 2006. A Method for Learning Part-Whole Relations. In *Proceedings of International Semantic Web Conference (ISWC)*, pages 723 – 735, Athens, GA, USA. Springer.

Ulrike von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17:395–416.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising Measures of Lexical Distributional Similarity. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1015–1021, Geneva, Switzerland.

Dominic Widdows and Beate Dorow. 2002. A Graph Model for Unsupervised Lexical Acquisition. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1093–1099, Taipei, Taiwan.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL)*, pages 197–204, Edmonton, Alberta, Canada.

Michael Wiegand, Benjamin Roth, and Dietrich Klakow. 2012a. Knowledge Acquisition with Natural Language Processing in the Food Domain: Potential and Challenges. In *Proceedings of the ECAI-Workshop on Cooking with Computers (CWC)*, pages 46–51, Montpellier, France.

Michael Wiegand, Benjamin Roth, and Dietrich Klakow. 2012b. Web-based Relation Extraction for the Food Domain. In *Proceedings of the International Conference on Applications of Natural Language Processing to Information Systems (NLDB)*, pages 222–227, Groningen, the Netherlands. Springer.

Michael Wiegand, Benjamin Roth, Eva Lasarcyk, Stephanie Köser, and Dietrich Klakow. 2012c. A Gold Standard for Relation Extraction in the Food Domain. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 507–514, Istanbul, Turkey.

Ichiro Yamada, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, Masaki Murata, Stijn De Saeger, Francis Bond, and Asuka Sumida. 2009. Hypernym Discovery Based on Distributional Similarity and Hierarchical Structures. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 929–927, Singapore.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with Local and Global Consistency. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver and Whistler, British Columbia, Canada.