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 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
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.

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

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

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

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
<th>Examples</th>
<th>Perc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISH</td>
<td>composite food items</td>
<td>cake, falafel, meat loaf</td>
<td>32.10</td>
</tr>
<tr>
<td>ATOM</td>
<td>non-composite food items</td>
<td>apple, steak, potato</td>
<td>67.90</td>
</tr>
</tbody>
</table>

Table 2: Distribution of dishes and atomic food items among the food vocabulary (*gold standard*).
Table 3: Domain-independent patterns for building the similarity graph.

Figure 1: Illustration of the similarity graph.

Table 4: List of patterns to extract seeds.

\begin{itemize}
\item \textbf{food\_item}, (or rather instead of "I") \textbf{food\_item}.
\end{itemize}

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 \( W_{ij} = \frac{C_{ij}}{d_i d_j} \) if \( i \neq j \), and \( W_{ii} = 0 \). The normalization weight \( \lambda \) 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 \( x_{i,1 \leq i \leq n} \) labeled as \( y_i \in \mathcal{L} \) and \( x_{i,1 \leq i \leq n} \) 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 \( \hat{y}_i = \arg \max_{y \leq c} F_i \). Conversely, the gold labeling matrix \( Y \) is a \( n \times c \) matrix with \( Y_{ij} = 1 \) if \( x_i \) is labeled as \( y = j \) and \( Y_{ij} = 0 \) otherwise.

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

\[ Q = \frac{1}{2} \left( \sum_{i,j=1}^n W_{ij} \left( \frac{\delta_{y_i, y_j}}{\sqrt{d_i d_j}} \right) + \mu \sum_{i=1}^n \| F_i - Y_i \| \right) \]

where \( \delta_i \) is the degree function of \( W \).

The first term in \( Q \) 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.

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 (\textit{fruit, meat, sweets}) represent the hypernyms for which we extract seed hyponyms (\textit{banana, beef, chocolate}).

In order to extract seeds for Task II, we apply two domain-specific sets of patterns (\textit{patt\_dish} and \textit{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:

\[ \text{score}(\text{food item}) = \#\text{patt\_dish}(\text{food item}) - \#\text{patt\_atom}(\text{food item}) \]

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 medi-
ate hyperonym 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 a few 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 \( L = I - D^{1/2} W D^{1/2} \), 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.

²Unlike 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.

³That is, \( D_{ii} \) equals to the sum of the \( i \)th row.

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-Topn) as seeds for graph-based optimization. The table shows that the semi-supervised graph-based approach with these seeds outperforms the baseline 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-Topn 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

⁴All manually labeled resources are available at: www.lsv.uni-saarland.de/personalPages/michael/reifood.html
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_{hears} (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.
Table 8: Comparison of different seed initializations for the food type categorization task (underlined food items represent erroneously extracted food items).

<table>
<thead>
<tr>
<th>Class</th>
<th>5 Manually Chosen Seeds (5-PROTO)</th>
<th>5 Hearst-Pattern Seeds (PAT-Top5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAT</td>
<td>schnitzel, rissale, bologna, redfish, trout</td>
<td>salmon, beef, chicken, turkey leg, poultry</td>
</tr>
<tr>
<td>BEVERAGE</td>
<td>coffee, tea, water, beer, coke</td>
<td>coffee, beer, mineral water, lemonade, tea</td>
</tr>
<tr>
<td>VEGETABLE</td>
<td>peas, green salad, tomato, cauliflower, carrot</td>
<td>zucchini, lamb's salad, broccoli, leek, cauliflower</td>
</tr>
<tr>
<td>SWEET</td>
<td>chocolate, forfe, popcon, apple pie, potato crisps</td>
<td>wine gum, marzipan, custard, pancake, biscuits</td>
</tr>
<tr>
<td>SPICE</td>
<td>pepper, cinnamon, salt, gravy, remoulade</td>
<td>cinnamon, laurel, clove, tomato sauce, basil</td>
</tr>
<tr>
<td>STARCH</td>
<td>spaghetti, basmati rice, white bread, potato, french fries</td>
<td>gratin potatoes, jacket potato, potato, potato, jam</td>
</tr>
<tr>
<td>MILK</td>
<td>yoghurt, gouda, cream cheese, cream, butter milk</td>
<td>butter milk, bovine milk, sour milk, goat cheese, sour cream</td>
</tr>
<tr>
<td>FRUIT</td>
<td>banana, apple, strawberry, apricot, orange</td>
<td>banana, strawberries, pear, melon, kiwi</td>
</tr>
<tr>
<td>GRAIN</td>
<td>hazelnut, pumpkin seed, rice, fouro, semolina, wheat</td>
<td>sesame, spelt, wheat, millet, barley</td>
</tr>
<tr>
<td>FAT</td>
<td>margarine, lard, colza oil, spread, butter</td>
<td>margarine, lard, resolified butter, coconut oil, tartar</td>
</tr>
<tr>
<td>EGG</td>
<td>scrambled eggs, fried eggs, chicken egg, omelet, pickled egg</td>
<td>eggs, fried eggs, albumen, offal, easter egg</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Class</th>
<th>10 Seeds Extracted with Patterns (PAT-Top10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISH</td>
<td>marzipan, four, potato, olive oil, water, sugar, cream, chocolate, milk, tomato</td>
</tr>
</tbody>
</table>

Table 10: Impact of the similarity representation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Similarity</th>
<th>Classifier</th>
<th>Acc</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food Type</td>
<td>distributional</td>
<td>nearest neighbour</td>
<td>53.4</td>
<td>51.1</td>
</tr>
<tr>
<td></td>
<td>distributional</td>
<td>graph</td>
<td>25.6</td>
<td>25.6</td>
</tr>
<tr>
<td></td>
<td>pattern-based</td>
<td>graph</td>
<td>80.2</td>
<td>77.7</td>
</tr>
<tr>
<td>Dish</td>
<td>distributional</td>
<td>nearest neighbour</td>
<td>76.8</td>
<td>75.2</td>
</tr>
<tr>
<td></td>
<td>distributional</td>
<td>graph</td>
<td>71.5</td>
<td>71.2</td>
</tr>
<tr>
<td></td>
<td>pattern-based</td>
<td>graph</td>
<td>83.0</td>
<td>80.1</td>
</tr>
</tbody>
</table>

4.1.5 Do we need a domain-specific corpus?

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.

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 SVMlight 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).

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.

1. I very often eat fish and chips. (Relation: SuitsTo)
2. 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
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.

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

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

Table 13: Description of the feature set.

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

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

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


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.


