Automated Construction of a Large Semantic Network of Related Terms for Domain-Specific Modeling

Henning Agt and Ralf-Detlef Kutsche

Database Systems and Information Management Group DIMA
Technische Universität Berlin
Einsteinufer 17, 10587 Berlin, Germany
{henning.agt,ralf-detlef.kutsche}@tu-berlin.de

Abstract. In order to support the domain modeling process in model-based software development, we automatically create large networks of semantically related terms from natural language. Using part-of-speech tagging, lexical patterns and co-occurrence analysis, and several semantic improvement algorithms, we construct SemNet, a network of approximately 2.7 million single and multi-word terms and 37 million relations denoting the degree of semantic relatedness. This paper gives a comprehensive description of the construction of SemNet, provides examples of the analysis process and compares it to other knowledge bases. We demonstrate the application of the network within the Eclipse/Ecore modeling tools by adding semantically enhanced class name autocompletion and other semantic support facilities like concept similarity.

Keywords: Domain-Specific Modeling, Terminology Extraction, N-Gram, Lexical Pattern, Semantic Relatedness.

1 Introduction

1.1 Motivation

Our research work is motivated by the goal to provide automated modeling support for model-driven software engineering (MDE). Particularly, we address domain-specific modeling [1], an approach to capture domain knowledge and notation in specialized programming and modeling languages tailored to a specific domain. These domain-specific languages (DSLs) [2] enable domain experts to participate in software development and facilitate automation of software system construction.

We focus on early phases of DSL development in which the problem domain is identified and domain knowledge is gathered [3]. It is usually during that phase that domain models in UML class diagram notation, metamodels (abstract syntax models) for DSLs, or entity-relationship diagrams for data-driven applications are created. All approaches have in common that a relatively simple

C. Salinesi, M.C. Norrie, and O. Pastor (Eds.): CAiSE 2013, LNCS 7908, pp. 610–625, 2013.

meta-language is used and conceptual structures of a domain are expressed using its terminology in order to improve the understanding of the problem field [4].

Our vision of semantic modeling support [5] is as follows: The content of a domain model is analyzed during development. Based on the terms used in the model the modeler receives suggestions on what he or she might include in the model (e.g., related classes, possible sub- or super-classes, attributes, aggregations). The suggestions are adapted each time the model is changed.

In this paper, we address the following challenge: *Given a set of terms in a domain-specific model, can we automatically identify a corresponding set of semantically related terms for this model, and rank them by relevance?* In order to achieve this kind of support, we investigate how domain-specific modeling can benefit from computational linguistics and knowledge-based methods.

1.2 Domain-Specific Modeling and Computational Linguistics

Working on the connection of different research areas, we briefly introduce the most important concepts in those fields relevant to this paper.

The main goal of this work is the support of the creation of **domain models**. They contain "concepts, terms and relationships that reflect domain insight" [6]. Our main concern is technical **terminology**. Terms are parts of specialized vocabularies and can be composed of single or multiple words.

As it is very difficult to find sufficiently large knowledge bases for domain modeling, we have to construct them ourselves by information extraction. Our work relies on word **n-grams** [7] and their frequencies in text corpora. An n-gram is a sequence of n consecutive words. The frequency of an n-gram is determined by counting all its occurrences in a given text collection. N-gram statistics are usually used in speech recognition and natural language processing to predict which word follows another word using probability of occurrence. We use the frequency to derive the degree of relatedness between terms.

We apply **part-of-speech (POS) tagging** [8], a natural language processing step in which the corresponding lexical category (e.g., noun, adjective) is assigned to each text token using the Penn Treebank tagset [9], e.g., *researchers/NNS* means, that the word is a plural noun. In this paper we use POS-tagging to identify technical terms.

Semantic relatedness [10] measures the degree of relationship between words or concepts. The relatedness can either be expressed as an explicit lexical or semantic relationship, such as hyponymy (e.g., a surgeon *isA* doctor), or as a numeric value within a certain scale. Semantic relatedness covers any kind of lexical or functional relation between words in contrast to semantic similarity, which only measures how similar two words are.

1.3 Contributions and Outline

To achieve our intended semantic modeling support with automated model element suggestions, we consider the following: We require a dictionary of terms that is big enough to cover a large portion of domains with all possible terms

that are used in those domains. The terms should be interconnected if they are semantically related, thus constituting a semantic network. The degree of relatedness should be quantified to enable ranking of related terms. The network should allow for retrieving related terms of a single query term and of multiple terms contained in a domain-specific model.

Figure 1 gives an overview of our approach. (1) It relies on automated text analysis to extract information about technical terms and their relatedness. The input is a large text corpus from which word and word sequence frequencies (n-gram statistics) are determined. (2) In our current work, we do not create the n-grams ourselves, but we use an existing n-gram dataset that was derived from a 360 billion English word corpus. First, we transform the n-gram statistics into a queryable database. (3) Then, all n-grams are tagged according to their part-of-speech and (4) all words are normalized using several rules. Both the tags and the normalized n-grams are stored in a database as well. (5) Based on syntactic patterns we perform a terminology co-occurrence analysis to derive semantically related terms. Using the co-occurrence frequencies we create a large-scale graph of terms with weighted edges denoting the degree of relatedness. (6) An interface to SemNet is provided to query for terms and retrieve ranked sets of related terms.

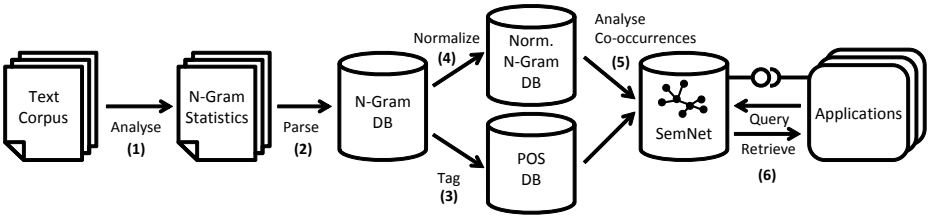


Fig. 1. Procedure of creating a semantic terminology network based on natural language statistics

The rest of this paper is organized as follows: In Section 2 we describe in detail how terminology and relatedness information is extracted from the Google Books n-gram dataset and how the semantic network is created. Section 3 shows the application of SemNet in a domain-specific modeling environment and provides examples of the content of the semantic network. In Section 4 we compare SemNet to existing semantic knowledge bases. Related work is given in Section 5, and Section 6 concludes the paper and describes future work directions.

2 Identifying Semantically Related Terms

In this section we first describe the properties of the input n-gram dataset and the kind of preprocessing that is required to extract information from such large data. We then illustrate how to identify terminology using lexical patterns and how the patterns are applied to find co-occurring terms. Finally, we show how to deduce probabilistic relationships between terms and how the semantic network is created.

2.1 The Google Books N-Gram Dataset

The Google Books project aims at providing a searchable digital library of a huge amount of books. Since 2004, Google Inc. digitized over 15 million books [11] for full text book search on the web using optical character recognition. Most of the books are provided by university libraries or publishers who participate in the partner programs.

Google selected a subset of approximately 5 million books of the years 1500 until 2008 and built a text corpus of roughly 500 billion words in several languages for quantitative text analysis. An n-gram analysis was performed that counts how often a certain word or word sequence occurs within the corpus. The resulting dataset includes word frequencies for all 1,2,3,4 and 5-grams that occurred at least 40 times[1].

The dataset is split into languages, and can be downloaded[2] as tab-separated plain text files. In our work, we use the English dataset (googlebooks-eng-all-20120701) that was derived from the English corpus (approximately 360 billion words in total). The dataset is 2.5 terabytes in size (1-grams and 5-grams) and contains over 61 billion lines of text. The structure of the files is given as follows.

```
n-gram TAB year TAB match_count TAB volume_count

the doctor and the patient      2002    281     216
the doctor and the patient      2003    262     205
```

For example, the first line of the 5-grams denotes that in the year 2002 the sequence *"the doctor and the patient"* occurred 281 times in 216 different books. We decided to use this dataset because it covers an extremely large variety of literature and terminology in almost every domain.

2.2 Preprocessing

Database Creation. Given 2.5 Terabytes of plain text input data, we first need to transform the n-gram data into a format that allows us to query and process it in reasonable time. The n-gram text files are parsed and stored in a relational database. In order to minimize memory requirements, the schema is kept simple, we store the complete vocabulary (1-grams) in one table and use foreign key relationships in the 5-gram table. During the complete process database creation it is kept in memory to reduce disk I/O, thus optimizing the processing time. The complete vocabulary consists of more than 10 million words/tokens and the database contains roughly 710 million 5-grams (21 GB data, 47 GB indices).

[1] The n-gram frequencies are separated by years of publication, but we only use the aggregated values. Evolution of words over time can be explored under `http://books.google.com/ngrams`

[2] The dataset can be downloaded at `http://books.google.com/ngrams/datasets`

Part-Of-Speech Tagging. For further terminology analysis, we use 5-grams only, because they provide the largest available context. We perform part-of-speech tagging for each of the 710 million 5-grams using the Stanford Log-linear Part-Of-Speech Tagger V3.1.3 [8] and store the tags in a database, too. The tagger assigns the lexical class to each word using the Penn Treebank tagset [9]. It operates context-sensitively with high accuracy and is able to identify the correct lexical class for ambiguous words that belong to multiple classes (e.g., the word *patient* can be a noun or an adjective). The newest version of the Google Books n-gram dataset already includes syntactic annotations. We did not use them because they are based on a cross-language tagset that does not allow the identification of proper nouns (see Section 2.3).

Normalization. Word variations are unified in the last preprocessing step. We perform plural stemming on all nouns (e.g., *doctors* → *doctor*) using the previously obtained part-of-speech information. Genitive 's is removed, and normal nouns and adjectives are lowercased. Figure 2 shows examples of normalized n-grams containing the word *doctor* with their part-of-speech tags.

id	word1	word2	word3	word4	word5	frequency
1	for	the	degree	of	doctor	86,176
2	the	doctor	-	patient	relationship	38,931
3	the	honorary	degree	of	doctor	15,464
4	between	doctor	and	patient	.	7,697
5	the	doctor	and	the	nurse	6,720
6	your	doctor	or	pharmacist	.	2,654
7	doctor	and	other	medical	personnel	1,095
...						

id	pos1	pos2	pos3	pos4	pos5
1	IN	DT	NN	IN	NN
2	DT	NN	:	NN	NN
3	DT	JJ	NN	IN	NN
4	IN	NN	CC	NN	.
5	DT	NN	CC	DT	NN
6	PRP$	NN	CC	NN	.
7	NN	CC	JJ	JJ	NN
...					

Fig. 2. Examples of normalized 5-grams and their corresponding part-of-speech tags (710 million rows in total, 21 GB + 14 GB disc space without indices)

2.3 Lexical Patterns

In order to find multi-word terms in n-gram natural language fragments we use lexical patterns similar to the lexico-syntactic patterns by Hearst [12]. We analyzed several existing dictionaries and determined the most frequent part-of-speech patterns of technical terms. They are predominantly composed of simple noun, noun-noun and adjective-noun combinations (approx. 77 percent of the terms). We summarize the most important patterns used for the terminology extraction in Table 1.

Special Patterns. The table also includes some special patterns that are required because of the tokenization of the input n-gram data. Words with hyphens are split into separate tokens, thus we include patterns for those cases (e.g., *NN : NN*). Usually, these words would be treated as single nouns. Foreign word patterns (*FW)* are required to identify special medical or biological terminology

Table 1. Excerpt of the lexical patterns of technical terms used in the analysis process (in decreasing order of frequency; 20 patterns in total)

Pattern	Explanation	Example
NN	Noun	the **doctor** and the **nurse**
JJ NN	Adjective-Noun Combination	**medical doctor** or a psychiatrist
NN NN	Noun-Noun Combination	**family doctor** for a checkup
NN : NN	Nouns with Hyphen	doctor or **nurse - midwife**
FW FW	Foreign Word Combination	doctor (**honoris causa**)
JJ NN NN	Adj-Noun-Noun Combination	doctor or **mental health professional**
SYM : NN	Hyphen Noun with Short Prefix	**co - operation** with doctor

that makes use of Latin words. Additionally to the patterns presented in Table 1, we allow several variations (e.g., *JJ NN NN*, *FW*, or *JJ : NN*). Please note that we explicitly exclude proper nouns because our main focus lies on conceptual terminology for domain-specific modeling.

Pattern Size. Currently, all our patterns have a size of three tokens at most. The reason for that is the limited context of a 5-gram. We can maximally identify a relationship between a single-word term and a triple-word term (see next section for more details). In our future work, we will derive our own n-gram statistics to be able to analyze a larger context with longer variations of the patterns. Nevertheless, the frequency of multi-word terms with four or more tokens is comparatively low.

2.4 Co-occurrence Analysis

The identification of semantically related terms is grounded in the *Distributional Hypothesis* first discovered by Harris [13] in the Fifties. It describes that words with similar meanings occur in similar contexts. In our case the context is a five word window given by a 5-gram. The absolute frequencies provide information on how often a specific context occurred. Consequently, terms that co-occur more often have a stronger relationship.

Stop Words. Prior to the analysis, we created a stop word list containing the most frequent words (e.g., "the", "of", "is", "to", "in", "a"), as well as punctuation and quotes, which are treated as separate tokens. We discard 5-grams that contain four or five stop words because they contain either one or zero terms. As a result, the size of the data is reduced to 58 percent and pattern matching is only applied to 415 million 5-grams.

Non-Consecutive Terms. In order to identify a semantic relation it is required to determine at least two co-occurring terms in one 5-gram. The terms must be separated by at least one token, for example by a coordinating conjunction (e.g., *and*, *or*), or by a preposition or subordinating conjunction (e.g, *of*, *in*,

for), or by special characters such as brackets. In comparison to Tandon et al. [14] who extract named relationships from n-grams, we explicitly exclude consecutive terms and include separation of terms by conjunctions because we want to extract the degree of relatedness between single- and multi-word terms.

Hierarchical Matching. Figure 3 shows examples of how the lexical patterns are hierarchically and non-consecutively applied. (a) Three simple nouns separated by a preposition and a conjunction are detected. (b) A single-word term and a three-word term are identified. The pattern on the highest level remains, respective lower level patterns are discarded. (c) This 5-gram actually contains a term consisting of four tokens which cannot be used to identify a semantic relation. The hierarchical pattern matching would detect the terms *part-time* and *doctor*. The 5-gram is discarded because they are in sequence. (d) In the English language certain nouns occur in almost every context because of their idiomatic use. Popular examples are: *number, part, kind, time, day*. We built a list of bad phrase patterns to exclude those occurrences (e.g., *part of*, *this time*, *each day*).

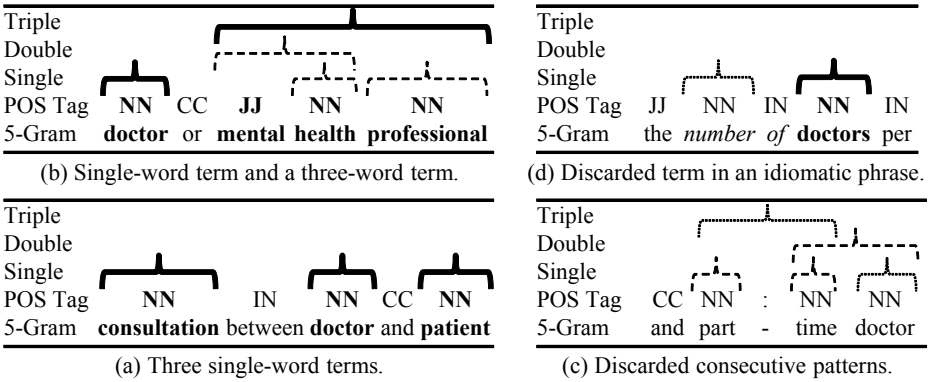
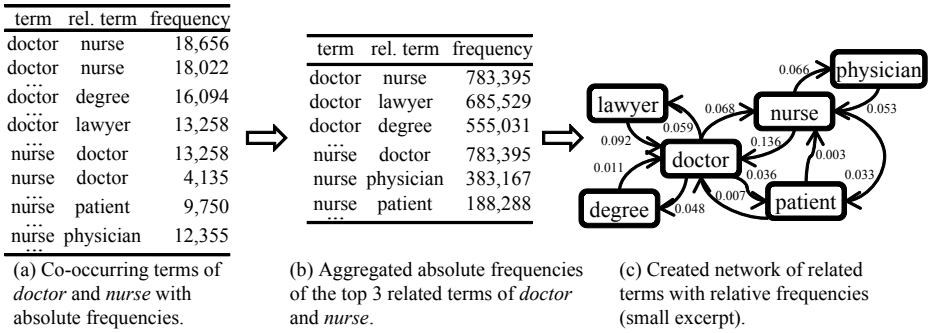


Fig. 3. Examples of the hierarchical application of lexical patterns

2.5 Network Creation

The result of the co-occurrence analysis is a large table of quantified binary relationships between terms (231.8 million relations in total, including back references). In fact, only 111.5 million 5-grams (15.69%) contained two or three terms. Figure 4a shows a small excerpt of the result. Since the same terms can co-occur in different 5-grams, the result contains many duplicates. As a next step we aggregate the absolute frequencies. Using the aggregated frequencies it is already possible to query for related terms ordered by strength of the relationship. Figure 4b shows the top 3 most related terms of *doctor* and *nurse*. In a last step we iterate through all terms and compute the relative frequency (co-occurrence probabilities) for each of its related terms with respect to the

other related terms. This normalization allows later comparison of the degree of relatedness across multiple terms (see Section 3.2). As a result we obtain a semantic graph in which each term is a node and each relationship is represented with two directed weighted edges as shown in Figure 4c. For example, the absolute frequency of *doctor* co-occurring with *nurse* equals that of *nurse* co-occurring with *doctor* (783,395 times). However, *doctor* also has strong relationships with *lawyer* and *degree* and additionally co-occurs with more terms than *nurse* does. Consequently, the relative frequency for *doctor*→*nurse* is lower than for *nurse*→*doctor*.



(a) Co-occurring terms of *doctor* and *nurse* with absolute frequencies.

(b) Aggregated absolute frequencies of the top 3 related terms of *doctor* and *nurse*.

(c) Created network of related terms with relative frequencies (small excerpt).

Fig. 4. Process of co-occurrence aggregation and relatedness degree computation

Properties of SemNet. The resulting network of semantically related terms comprises 2.7 million terms and 37.5 million weighted, directed edges. It requires 2.2 GB disk space, thus fitting into main memory on standard PC hardware. The automated analysis identified 268,937 distinct single-word terms, 2,115,494 double-word terms and 355,689 triple-word terms. We provide a relational database version (SQLite) and a graph database version (Neo4J) of SemNet for download and offer a web interface to query the network[3].

3 Application of SemNet

In this section we provide examples of SemNet, illustrate how joint semantically related terms are retrieved from the network for multiple input terms and describe an application of SemNet in a domain-specific modeling environment.

3.1 Querying Single Terms

Obtaining related terms for a single term from SemNet is a straightforward task. We developed a Java and PHP API for the network to retrieve ranked lists of related terms for given input terms. Terms can also be queried directly using

[3] `http://www.bizware.tu-berlin.de/semnet/`

SQL for the SQLite version or Cypher Query Language for the Neo4J version of SemNet. Table 2 shows examples of the 10 most related terms for terms of different degrees of specificity.

Table 2. Examples of the top 10 automatically identified related terms for terms with different degrees of specificity (f – absolute term frequency in the original text corpus, $\#r$ – number of related terms)

	teacher	doctor	electricity	software engineering	lymphocytic choriomeningitis
f	32.4M	19.1M	7.2M	212K	23K
1	student	nurse	water	CASE	virus
2	parent	lawyer	gas	field	LCM
3	school	degree	quantity	component	mouse
4	pupil	office	magnetism	area	cell
5	child	patient	heat	computer science	syngeneic
6	administrator	hospital	use	component	cytotoxicity
7	role	teacher	conductor	system	mediated cytotoxicity
8	training	order	current	discipline	mumps
9	work	law	steam	aspect	lymphocyte
10	principal	dentist	amount	term	monkey
$\#r$	8728	5519	2716	144	31

3.2 Querying Multiple Terms

For the usage of SemNet in a modeling tool it is not sufficient to retrieve related terms just for single terms. All terms in a model should be jointly considered.

Ranking Common Terms. We implemented the following strategy in our query interface to retrieve a set of related terms for multiple input terms: For each of the input terms we obtain the set of related terms together with their co-occurrence probabilities. All sets of related terms are intersected to determine a common set of related terms. In order to determine a new ranking of the common terms the co-occurrence probabilities are multiplied and decreasingly ordered. This ensures, for example, that a related term of high importance in one set and of less importance in another set will be ranked in a middle position in the joint result. In case n terms ($n > 2$) are queried, we repeat the intersection and probability computation for subsets of $n - 1$ input terms and rank the results after the very first intersection of all input terms. This avoids empty results in case many terms are queried but ensures that common semantically related terms are ranked higher.

Dealing with Ambiguity. Consequently, this mechanism allows to deal with ambiguity of terms. Imagine a query for *database* and *table*. Top most related term for *table* is *chair* in the sense of furniture. Second most related term for *table* is *contents* in the sense of a tabular array. A combination of the related terms of *database* and *table* as described before will lower the rank of all furniture related terms or exclude them.

3.3 Semantic Autocompletion in a Domain-Specific Modeling Tool

SemNet is used in the context of the research project BIZWARE[4], a collaboration of two academic partners and eight small and medium software enterprises. The industrial partners develop domain-specific languages (DSLs) in their respective business domains and the main task of the academic partners is the development of methods and tools to support DSL development.

A commonly used tool for DSL development and domain modeling is the Eclipse Ecore Diagram Editor[5]. We developed an extension for it, called Semantic Autocompletion (SemAcom) [15]. Whenever a new class is created in the diagram the developer can activate a context-sensitive pop-up list of related terms with a Ctrl-Space keystroke. The terms are retrieved from SemNet depending on the current content of the model. The suggestions are filtered while typing, thus providing a feature similar to autocompletion in search engines. Figure 5 shows SemAcom in action.

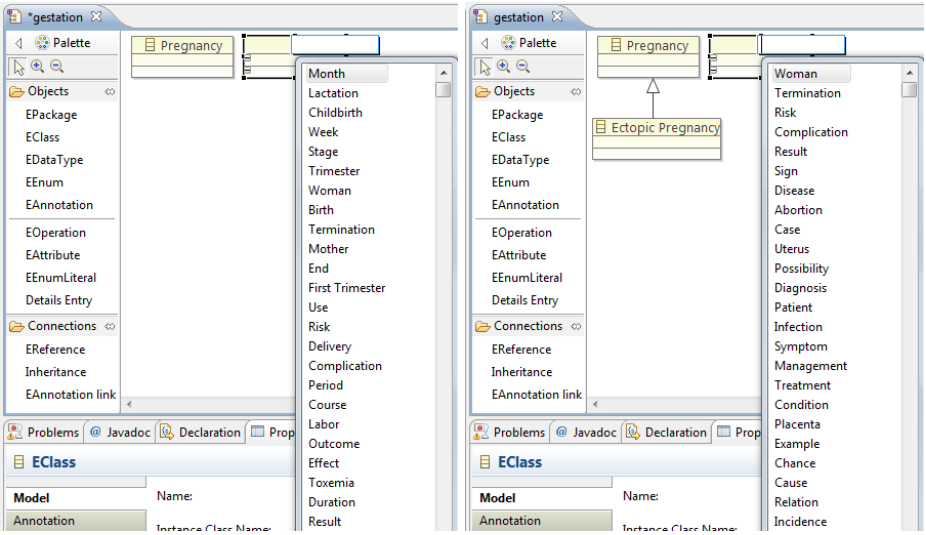


Fig. 5. Modeling with semantic autocompletion in the Ecore Diagram Editor. *Left*: SemAcom provides suggestions for the term "Pregnancy". *Right*: The suggestions are adapted according to the newly created class "Ectopic Pregnancy".

4 Comparison to Other Semantic Knowledge Bases

The evaluation of information extraction techniques is difficult because gold standards only exist for a few subtasks of it. We apply the following strategy to

[4] This work is partially supported by the Bundesministerium für Bildung und Forschung BMBF under grant number 03WKBU01A.

[5] `http://www.eclipse.org/modeling/emft/?project=ecoretools`

assess the content of SemNet: We compare it to two existing (partially) manually created semantic databases: WordNet V3.0 [16] and ConceptNet V5.1 [17]. We chose these two works for the following reasons. On the one hand, they contain information on terminology and their semantic relations, similar to SemNet. On the other hand, both projects focus on conceptual knowledge that can be used in the area of domain-specific modeling [18,19]. Automatically created knowledge bases such as YAGO[6] and DBpedia[7] have limited benefit for domain modeling because they concentrate on factual knowledge (on instance level).

Using the concrete example *pregnancy*, we first show what kind of information is contained in the respective networks and how it is represented. Secondly, we compare how much information of WordNet and ConceptNet is contained in SemNet.

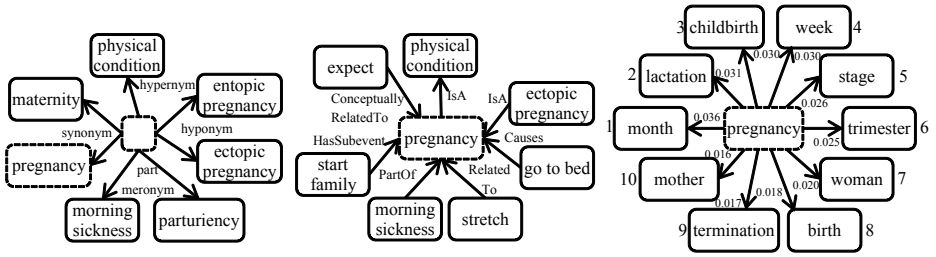
WordNet. WordNet is a lexical database for the English language [16]. It models synsets that group words sharing the same sense. It contains word senses for nouns, verbs, adjectives and adverbs (117,659 synsets in total, 82,115 nouns synsets and 102,249 noun relations). WordNet mainly comprises synonymous, taxonomic and part-whole relations. Figure 6a shows 7 out of 32 relations of the term *pregnancy*. The word sense in the middle groups the synonyms *pregnancy* and *maternity* and relates them to other senses.

ConceptNet. ConceptNet is a "large semantic graph that describes general human knowledge" [17]. It models concepts that are expressed in natural language phrases. It was created manually based on the Open Mind Common Sense project[8] and partially automatically from Wiktionary and the ReVerb project. Lexical types are not differentiated, it contains concepts such as *database software*, *beautiful*, and *build aircraft* (414,188 English concepts in the core version of it and 903,621 relations between them). Besides taxonomic and part-whole relations it contains several other relation types (e.g., *AtLocation*, *HasProperty*). Figure 6b shows examples (7 out of 58 relations) for the concept *pregnancy*.

SemNet. In SemNet we automatically created a graph of noun terminology (2,740,120 terms). Edges between terms are probabilistic links that represent the latent semantic association between words based on the Distributional Hypothesis [13] (37,542,622 relations). Figure 6c shows the term *pregnancy* together with its 10 most related terms (4,039 relations in total, for space reasons we omit back references).

Quantitative Evaluation Procedure. The evaluation of how much information of WordNet and ConceptNet is contained in SemNet is performed in two steps. We first determine how much of WordNet's and ConceptNet's noun terminology is included in SemNet. Secondly, we take the found synsets and concepts,

[6] `http://www.yago-knowledge.org`
[7] `http://dbpedia.org`
[8] `http://csc.media.mit.edu/`



(a) The word sense *pregnancy* in WordNet (7 out of 32 relations).

(b) The concept *pregnancy* in ConceptNet (7 out of 58 relations).

(c) The term *pregnancy* in SemNet (First 10 out of 4039 relations).

Fig. 6. Examples of how terminology information for the topic *pregnancy* is represented in WordNet, ConceptNet and SemNet

respectively, and determine how many of their relations are contained in SemNet. Therefore, we can examine how well the specific relations can be detected by our applied methods of statistical semantics.

Nouns are already classified in WordNet (82,115 synsets), we exclude 7,714 instance synsets (e.g., city names) and synsets solely having terms with more than 3 words (1122) or only containing digits and special characters (285). As a result 72,994 noun synsets are evaluated. A synset is counted as found if at least one of the synonyms is contained in SemNet.

The comparison of ConceptNet with SemNet is challenging because concept names in ConceptNet can contain all types of lexical expressions, thus we cannot select all nouns. We tried to determine noun concepts by POS tagging, but it is too imprecise for single words with no context. Because of that we determined all concepts in ConceptNet that are also nouns in WordNet. As a result 49,301 concepts are evaluated.

Relations of WordNet and ConceptNet are evaluated as follows. All found synsets/concepts are iterated and for each of them we retrieve the related terms from SemNet. We then determine how many of the WordNet/ConceptNet relation targets are contained in the list of related terms of SemNet.

Quantitative Evaluation Results. 56,321 of 72,994 noun synsets in WordNet (77,16%) have been found in SemNet. The 72,994 synsets comprise 98,681 distinct noun terms of which 61,349 (62,17%) have been found in SemNet. Noun coverage is relatively low compared to synset coverage because in many cases one synonym of the synset is found in SemNet but other rare synonyms cannot be found due to the threshold of at least 40 occurrences in the n-gram dataset (a 5-gram must occur at least 40 times in the original text corpus to be included in the n-gram dataset). This threshold cannot be relaxed at the moment because of the way the Google Books n-gram dataset is constructed and distributed. 40,625 of 49,301 concepts in ConceptNet (82,40%) have been found in SemNet.

The results of the relation analysis are summarized in Table 3. 61,931 explicit hyponym/hypernym and meronym/holonym relations and 11,832 implicit synonym relations of WordNet have been evaluated. 256,213 relations of

Table 3. Results of the relation evaluation

WordNet			ConceptNet		
Relation Type	Number of Relations	SemNet Coverage	Relation Type	Number of Relations	SemNet Coverage
hyponym/hypernym	53,785	48,53%	IsA	90793	45,85%
			RelatedTo	21936	69,66%
			AtLocation	19408	53,86%
			HasProperty	19265	63,25%
			have or involve	16101	74,78%
			ConceptuallyRelatedTo	11166	75,42%
			UsedFor	9313	66,53%
			HasA	7829	80,65%
meronym/holonym	8,146	46,03%	PartOf	5914	48,92%
synonym	11,832	52,87%	SimilarTo	1467	25,77%

ConceptNet have been evaluated (for space reasons we only include the most frequent relations of ConceptNet).

The very good results for `RelatedTo` / `ConceptuallyRelatedTo` relations support that our methods accomplish the identification of semantically related terms. Average results are achieved for taxonomic and part-whole relations. The biggest coverage is gained for membership relations (`have or involve`, `HasA`) because the distance between two terms in natural language expressions indicating such a relationship is low. Thus, 5-grams include them more often. Similar observations have been made by Nulty et al. [20].

Additionally to the explicit relations, we also compared WordNet's synonym relations, implicitly given by the synset, with SemNet. For example, given the synset *(nanny, nursemaid, nurse)*, we evaluate the relations *nanny*↔*nursemaid*, *nursemaid*↔*nurse*, and *nanny*↔*nurse*. 52,87% of the 11,832 evaluated synonym relations are contained in SemNet. In ConceptNet, `SimilarTo` is the only relationship indicating synonymy. SemNet only covers 25,77% of these relations. The reason for that is that the identification of these relationships usually require sentence level analysis [12,14] which is not possible with 5-grams.

In summary, the automated identification of semantically related terms shows very good results, although only a context of five words given by a 5-gram is available. Compared to manually created knowledge bases with a few hundred thousand terms and relations, SemNet comprises a variety many times greater.

5 Related Work

Since this work is related to several research areas, we summarize the most important approaches in the following categories:

Automated Construction of Semantic Knowledge Bases. Research on extracting information from semi- and unstructured data sources has especially been boosted by Semantic Web and Linked Open Data initiatives in the last decade. Popular examples of automatically constructed knowledge bases are YAGO and DBpedia that extract instance knowledge from Wikipedia, but we

focus on the extraction of conceptual knowledge. Similar to our approach, **n-grams** are analyzed by Tandon et al. [14]. Their focus is the population of ConceptNet by learning patterns for specific relations limited to single-word terms. In contrast we use the n-grams to extract semantically related multi-word terms. **Terminology extraction** is mainly investigated in the area of document-based information retrieval. The baseline model in this area is tf-idf. State-of-the-art systems use supervised learning or graph-based methods and external knowledge sources [21]. Because of the small n-gram language fragments we rely on part-of-speech patterns, similar to lexico-syntactic patterns by Hearst [12]. Nulty et al. [20] also investigate lexical patterns in n-grams, but concentrate on the patterns that separate the terms. We deduce **semantic relatedness** between terms based on statistical semantics [22] using n-gram frequencies. Alternative approaches use wikipedia-based explicit semantic analysis [23] or combine WordNet concept hierarchies and collaboratively constructed knowledge sources [10].

Modeling Support by Semantic Knowledge Sources. Semantic modeling support has been predominantly investigated in the area of connecting **ontology** development and model-driven development [24] and model reuse. Tairas et al. [18] describe how the domain analysis phase of DSL development benefits from the use of ontologies. Their approach is based on manual ontology construction during early stages of domain-specific language development. Thonggoom et al. [25] support conceptual modeling using data model instance repositories. The repositories are created from SQL schema libraries with several hundred relations, thus containing patterns from prior database designs to enable **modeling knowledge reuse**. The REBUILDER UML system [26] aims at a similar goal for UML diagram reuse. The design assistant uses case-based reasoning. Both approaches are comparable to our semantic autocompletion application. In contrast to our solution they can suggest model fragments, but are dependent on the relatively small size of the input data.

6 Conclusion and Future Work

We presented an approach to automatically extracting multi-word terms and their degree of semantic relatedness from n-gram natural language statistics. Using only a window of five words given by the 5-grams and 20 lexical patterns, we have been able to create SemNet, a graph of related terms with 2.7 million nodes and 37.5 million probabilistic edges denoting the latent semantic relationship between them. We demonstrated the usage of the semantic network in a domain-specific modeling environment providing semantically enhanced class name autocompletion for the Ecore Diagram Editor. However, the usage of SemNet is not limited to modeling. For example, it can be used for keyword expansion in search, for automated topic suggestions [27] or as background knowledge for natural language processing tasks.

In our future work, we will derive our own n-gram statistics from text corpora in order to analyze a larger context and to remove the limitation that terms

consist of three words at most. We will apply our methods to other languages, especially for German we expect better term coverage because of the more frequent use of compounds. Currently, we investigate how to effectively combine probabilistic information with specific relations in knowledge bases. The semantic network itself leaves plenty of room for applying clustering in order to find domains. Finally, we plan to implement more types of modeling suggestions (e.g., attributes, relations, abstractions/refinements) and even complete model fragments by investigating patterns in existing domain models.

References

1. Kelly, S., Tolvanen, J.P.: Domain-Specific Modeling: Enabling Full Code Generation. Wiley-IEEE Computer Society Press (March 2008)
2. Fowler, M.: Domain Specific Languages. Addison-Wesley, Boston (2010)
3. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Comput. Surv. 37, 316–344 (2005)
4. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer-Verlag New York, Inc., Secaucus (2007)
5. Agt, H.: Supporting Software Language Engineering by Automated Domain Knowledge Acquisition. In: Kienzle, J. (ed.) MODELS 2011 Workshops. LNCS, vol. 7167, pp. 4–11. Springer, Heidelberg (2012)
6. Evans: Domain-Driven Design: Tacking Complexity in the Heart of Software. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
7. Jurafsky, D., Martin, J.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Prentice Hall series in artificial intelligence. Prentice Hall (2000)
8. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of the NAACL 2003, pp. 173–180. Association for Computational Linguistics, Stroudsburg (2003)
9. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)
10. Zesch, T.: Study of Semantic Relatedness of Words Using Collaboratively Constructed Semantic Resources. PhD thesis, TU Darmstadt (February 2010)
11. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Team, T.G.B., Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A., Aiden, E.L.: Quantitative Analysis of Culture Using Millions of Digitized Books. Science 331(6014), 176–182 (2011)
12. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Conference on Computational Linguistics, COLING 1992, Stroudsburg, PA, USA, vol. 2 (1992)
13. Harris, Z.: Distributional structure. Word 10(23), 146–162 (1954)
14. Tandon, N., de Melo, G., Weikum, G.: Deriving a Web-Scale Common Sense Fact Database. In: AAAI (2011)
15. Agt, H.: SemAcom: A System for Modeling with Semantic Autocompletion. In: Model Driven Engineering Languages and Systems - 15th International Conference, MODELS 2012, Demo Track, Innsbruck, Austria (2012)

16. Fellbaum, C.: WordNet: An Electronic Lexical Database. The MIT Press, Cambridge (1998)
17. Speer, R., Havasi, C.: Representing General Relational Knowledge in ConceptNet 5. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey (2012)
18. Tairas, R., Mernik, M., Gray, J.: Using Ontologies in the Domain Analysis of Domain-Specific Languages. In: Chaudron, M.R.V. (ed.) MODELS 2008. LNCS, vol. 5421, pp. 332–342. Springer, Heidelberg (2009)
19. Agt, H., Kutsche, R.D., Wegeler, T.: Guidance for Domain Specific Modeling in Small and Medium Enterprises. In: SPLASH 2011 Workshops. Proceedings of the Compilation of the Co-located Workshops on DSM 2011, Portland, OR, USA (2011)
20. Nulty, P., Costello, F.: Using lexical patterns in the Google Web 1T corpus to deduce semantic relations between nouns. In: Proceedings of the Workshop on Semantic Evaluations, DEW 2009, Stroudsburg, PA, USA, pp. 58–63 (2009)
21. Grineva, M., Grinev, M., Lizorkin, D.: Extracting key terms from noisy and multitheme documents. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, pp. 661–670. ACM, New York (2009)
22. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Int. Res. 37(1), 141–188 (2010)
23. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI 2007, San Francisco, CA, USA (2007)
24. Henderson-Sellers, B.: Bridging metamodels and ontologies in software engineering. J. Syst. Softw. 84, 301–313 (2011)
25. Thonggoom, O., Song, I.-Y., An, Y.: Semi-automatic conceptual data modeling using entity and relationship instance repositories. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 219–232. Springer, Heidelberg (2011)
26. Gomes, P., Gandola, P., Cordeiro, J.: Helping software engineers reusing UML class diagrams. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 449–462. Springer, Heidelberg (2007)
27. West, R., Precup, D., Pineau, J.: Automatically suggesting topics for augmenting text documents. In: Proceedings of the 19th International Conference on Information and Knowledge Management, CIKM 2010. ACM, New York (2010)