Towards context-based disambiguation of mathematical expressions

Mihai Grigore[1] and Magdalena Wolska[2]* and Michael Kohlhase[1]

[1] School of Engineering & Science, Jacobs University Bremen, D-28759 Bremen, Germany
{m.grigore,m.kohlhase}@jacobs-university.de

[2] Fachrichtung 4.7 Allgemeine Linguistik, Universität des Saarlandes
66 041 Saarbrücken, Germany
magda@coli.uni-sb.de

Abstract

We present a preliminary study on disambiguation of symbolic expressions in mathematical documents. We propose to use the natural language within which the expressions are embedded to resolve their semantics. The approach is based on establishing a similarity between the expression's discourse context and a set of terms from *Term Clusters* based on OpenMath Content Dictionaries. The Term Clusters are semi-automatically constructed terminological resources which classify related mathematical concepts into groups. Each group is labelled with a term which represents the common denominator between the concepts.

1 Motivation

Technical and scientific documents have been gaining increasing attention in the computational linguistics community. These documents stretch the current natural language processing technology, among others because they contain embedded structures such as tables, diagrams, or mathematical formulae, which interact with the textual content. While interpretation of such structures is currently outside of the state-of-the-art in language processing, their automatic understanding will enable us to provide services such as fact search, plagiarism detection, and change management for technical and scientific documents. In this paper we address this problem from a linguistic perspective and present a step towards the semantics construction of mathematical formulae. Concretely, we address symbol overloading as one of the sources of ambiguities occurring in mathematical notations.

Authors tend to exploit established conventions in mathematical notation leaving some of the ambiguous notation without explicit explanation, relying on the reader being able to recover the intended meaning. Consider the expression "ω^{-1}": If ω is known to be a function, then ω^{-1} is the inverse function corresponding to ω. However, if ω is a scalar, ω^{-1} should be understood as $1/\omega$. Consider now the following expression: "$S^{-B}f(C \ln S)$". It is a complex term consisting of two subterms, S^{-B} and f, whose concatenation denotes multiplication: S^{-B} is a scalar by which f is multiplied. $f(C \ln S)$, in turn, is a function: here, concatenation of f and $(C \ln S)$ denotes function application. The different appropriate interpretations of the superscript and symbol concatenation are taken for granted by the reader.

*Correspondence to: Universität des Saarlandes, Allgemeine Linguistik (FR 4.7), Postfach 15 11 50, 66 041 Saarbrücken, Germany; Tel:+49-681-3024345 Fax:+49-681-3024351

Mathematical discourse, however, does not consist of symbolic expressions alone. It is rather the familiar combination of natural language and symbolic expressions. Crucially for interpretation, the expressions' linguistic context often contains information which helps determine the expressions' meaning. Readers can therefore in many cases immediately resolve the intended reading of mathematical expressions by looking at the embedding discourse context. For instance, in the second example the text actually reads:

> *"... The scaling function has the form $S^{-B}f(C \ln S)$, where f is a 2π-periodic function."* from [9]

Given this linguistic context, the symbolic expression can be immediately interpreted to denote a function and with the knowledge that S^{-B} is a scalar, the internal structure of the expression can be identified.

The intended interpretation of symbolic mathematical expressions can be a useful source of information in a number of sub-tasks in a mathematical document processing pipeline for digitalising mathematics. For instance, in the task of parsing mathematical notation, i.e. identifying the structure and (compositional) semantics of symbolic expressions, the information about the expressions' interpretation can guide the selection (or weighing) of likely parse candidates. This could be useful in processing LaTeX documents as well as in mathematical OCR, in particular, in handwriting recognition; for instance, in examples such as above, in deciding between horizontal adjacency and super-/subscript relation when the super-/subscript is written partly across the centre horizontal line of the expression.

In this paper we present a preliminary study on disambiguating a certain subset of symbolic expressions in mathematical documents. Namely, we focus on those mathematical expressions which are syntactically part of a *nominal group* and, in particular, are in an *apposition relation* with an immediately preceding noun phrase. That is, our target expressions come from a linguistic pattern: "*... noun_phrase symbolic_math_expression ...*" (as in the example above). Therefore, in the approach described here we assume that a target mathematical expression can be disambiguated using its *left* context.

We formulate the disambiguation problem as follows: Given a mathematical document containing a target mathematical expression, can we indicate one (or more) concepts from a predefined set of concepts as the interpretation of the given expression. We claim that the linguistic context information improves disambiguation accuracy.

Our approach is based on the use of natural language lexical context information which is contained in the natural language surrounding a target expression. We compute a semantic similarity between the words from the lexical context of a given expression and a set of terms from manually constructed and semi-automatically extended *Term Clusters* based on OpenMath. Lexical contexts for the Term Clusters are compiled from a large corpus. As interpretation of a mathematical expression we consider the cluster (represented by its name) with the highest similarity to the words in the context.

Outline The paper is organised as follows: In Section 2 we describe our approach to disambiguation of appositional mathematical expressions: the lexical resources it uses and the word similarity-based disambiguation. In Section 3 we outline the experiment setup: the data we used, the performance metrics, and the baselines. The results are presented in Section 4. In Section 5 we present conclusions and discuss further work.

Symbol Name	CD Name (CD Group)	Description
`inverse`	`fns1` (Functional Operators)	This symbol is used to describe the inverse of its argument (a function). This inverse may only be partially defined because the function may not have been surjective. If the function is not surjective the inverse function is ill-defined without further stipulations. No assumptions are made on the semantics of this inverse.
`inverse`	`arith2` (Arithmetic Functions)	A unary operator which represents the inverse of an element of a set. This symbol could be used to represent additive or multiplicative inverses.
`eigenvector`	`linalg4` (Linear Algebra)	This symbol represents the eigenvector of a matrix. It takes two arguments the first should be the matrix, the second should be an index to specify which eigenvalue this eigenvector should be paired with. The ordering is as given in the eigenvalue symbol. A definition of eigenvector is given in Elementary Linear Algebra, Stanley I. Grossman in Definition 1 of chapter 6, page 533.

Table 1: Excerpt from OpenMath Content Dictionaries

2 The Approach

Our approach is based on two observations: First, the linguistic context is often a good indicator of the intended semantics of a mathematical expression. Second, similar lexical contexts indicate similar mathematical domain content. The central part of the approach is the use of co-occurrence statistics in computing semantic similarity between the words from the mathematical expressions' context and a set of terms from predefined semantic classes. These are represented by manually constructed Term Clusters which model certain logical classes of mathematical concepts according to lexical terms which evoke them.

2.1 Lexical Resources

The main resource on which our approach relies is a collection of *Term Clusters* (TC) which we derive from OpenMath *Content Dictionaries* (CDs). OpenMath [6, 14] is a language for representing mathematics, in particular symbolic mathematical expressions, both at the surface structure and the semantics level. It is becoming a de facto standard for communicating content-based mathematics over the Web. OpenMath uses CDs to define the semantics of symbols used to build mathematical expressions.

OpenMath Content Dictionaries The OpenMath CDs group symbol definitions by sub-fields of mathematics and carry names (e.g. arith2, linalg4, set1) that reflect this (see Table 1). We make use of the fact that they are further organized into groups according to mathematical areas (Arithmetic Functions, Linear Algebra, etc.). Note that mathematical concepts are identified by a symbol name (e.g. inverse, eigenvector, emptyset) together with CD names, so that CDs can have symbols of the same name, which are nonetheless different mathematical concepts. Note also that one CD can belong to more than one group.

TCN	Representative Terms
`algebraic structure`	algebra, array, basis, field, generator, group, groupoid, ideal, lattice, matroid, monoid, quaternion, ring, semigroup, space, subfield, submonoid, subsemigroup, ...
`function`	application, automorphism, closure, convolution, density, eigenfunction, endomorphism, entropy, function, functor, hamiltonian, hermitian, homomorphism, homotropy, inverse, isomorphism, lagrangian, logarithm, map, morphism, transformation, ...
`property`	associativity, commutativity, concavity, continuity, convexity, differentiability, diffusion, distributivity, goodness, linearity, noise, property, violation, ...

Table 2: Excerpt of the Term Clusters

Each symbol declaration in a CD is accompanied by a natural language description of the symbol's meaning. The descriptions are written according to an informal set of guidelines [8]. They share a certain formulaic style, relatively simple syntactic structure, and are characterized by brevity. There are 190 CDs in the currently available OpenMath [15]. Table 1 shows the definitions of an *eigenvector* and two definitions of the *inverse* concepts.

Term Clusters for Mathematical Expression Disambiguation We will use the OpenMath CDs to build Term Clusters as linguistic terminological resources that group mathematical lexical terms (corresponding to mathematical concepts) into sets of terms subordinate to a more general term (concept). It is these more general terms, Term Cluster Names (TCN), that we will use as interpretations of mathematical expressions in the disambiguation process: for each target expression we will assign a TCN as its interpretation.

For the experiments described in this paper we created 17 TCs semi-automatically. The structure and the content of the TCs were to a large extent inspired and semi-automatically extracted from the OpenMath Content Dictionaries.* The Term Clusters were constructed as follows: First, we extracted mathematical terms from each OpenMath CD and removed modifiers from multi-word terms obtaining bare nouns. Next, the CDs associated with the same mathematical area were collapsed (e.g. `arith1`, `arith2` and `arith3` correspond to `arithmetics`). The same procedure was applied to pairs of CDs with high term overlaps. We then extended the set of terms per cluster by extracting the top 100 most frequently co-occurring words from 10,000 documents from the arXMLiv collection [17]. Co-occurrence frequencies were calculated for pairs of words within one sentence.

Finally, in order to further enrich the lexical resource we partially automatically and partially manually extracted terms from various online lexica of mathematical terms, in particular, the University of Cambridge mathematical thesaurus [7] and the MathWorld lexicon of mathematical terms [11], and added them to the appropriate Term Clusters. The lexical entries in the TCs are bare nouns in the singular form. Table 2 shows excerpts from the TCs *algebraic structure*, *function* and *property*.

*In initial experiments, we used a pre-processed version of OpenMath CDs , but found its granularity too fine-grained on the one hand, and on the other hand, its coverage too limited.

2.2 Mathematical Expression Disambiguation

The method we propose is inspired by recent computational approaches to word sense disambiguation and lexical similarity which use statistical association measures to estimate semantic relatedness between words by analysing their distributional properties. In these approaches distributional similarity is computed based on, for instance, WordNet glosses [21] the Wikipedia, large corpora, or even the Web (see, for instance, [3, 12, 13, 16, 19]).

By analogy, our approach considers similarity between the given mathematical expression's lexical context and the terms in Term Clusters introduced above. The disambiguation process consists of three parts: First, we preprocess the documents and identify the candidate target mathematical expressions, then we compute corpus-based similarities for each TC, and finally disambiguate each target expression based on the words in their context and the TCs. Below we briefly present the disambiguation components.

Preprocessing We used 10,000 mathematical documents from the arXMLiv collection [17], word- and sentence-tokenized them, stemmed the words, and normalized the mathematical expressions by replacing them with a unique identifier. This preprocessing was performed both on the corpus for similarity computation and on the documents we used for evaluation of the approach. The latter documents were also processed with the Stanford part-of-speech (POS) tagger [18] in order to identify nouns in the left context of appositional mathematical expressions. The recall results we report in Section 4 are based on the output of this tagger.[†]

The evaluation set we furthermore processed in two ways in order to obtain results for alternative approaches to selection of candidates for computing semantic similarity: stop-word based and POS-based. In the stop-word approach the candidate terms for computing similarity were selected using a stop-word list alone. In the POS approach we used those words which were tagged as nouns by the Stanford tagger.

Computing term similarity We experimented with three statistics to find words semantically related to the TC terms: the Dice coefficient (*Dice*), the pointwise mutual information (*PMI*) and the z-score (z). These measures estimate relative probability with which words occur in proximity and have been previously successfully used in computational linguistics [4, 5, 20]. For two words w_1 and w_2, *Dice* is defined as twice the ratio of the joint probability to the sum of the individual probabilities, *PMI* is defined as the *log* of the ratio of the probability of the words occurring together to the product of the individual probabilities, and z is the proportion of the difference of the expected and the observed probabilities (P) to the expected probability (E):

$$Dice(w_1, w_2) = \frac{2 \times P(w_1, w_2)}{P(w_1) + P(w_2)}$$

$$PMI(w_1, w_2) = \log_2 \frac{P(w_1,w_2)}{P(w_1) \times P(w_2)} \qquad z(w_1, w_2) = \frac{P(w_1,w_2) - E(w_1,w_2)}{\sqrt{E(w_1,w_2)}}$$

Because *log* is monotonically increasing, relative ordinal rankings of *PMI* estimates are preserved if *log* is dropped. We approximate the probabilities by raw frequencies, as is a common practice.

[†]We are aware of the fact that these results are affected by the low performance of the tagger which was not trained on documents from our domain. To our knowledge, there are currently no available dedicated language processing tools, in particular, part-of-speech taggers, for mathematical discourse. In the future, we are planning to develop a dedicated tagger and re-evaluate our approach on more accurate POS outputs.

Based on experimentation we used the *Dice* coefficient for those words with *Dice* scores higher than a certain predefined threshold, $\lambda \approx 0.6$, otherwise we used either *PMI* or z:

$$sim(w_1, w_2) = \begin{cases} Dice(w_1, w_2) & \text{if } Dice(w_1, w_2) > \lambda \\ PMI(w_1, w_2) \text{ or } z(w_1, w_2) & \text{otherwise} \end{cases}$$

PMI is a frequently used statistical estimator of the strength of word co-occurrence; see, for example, [20]. The use of different lexical co-occurrence statistics is motivated by the fact that mutual information is reported to be less efficient on low-frequency events [10]. Comparisons with latent semantic analysis show that *PMI* achieves better results when large amounts of data is used [4, 5]. In Section 4 we report results for different similarity thresholds: for *PMI* we experimented with thresholds $\delta \in \{0.6, 0.8, 0.9\}$, while for z the considered values were $\delta \in \{0.0, 10.0, 20.0\}$ Lexical similarity was computed based on a subset of arXMLiv documents preprocessed as described above. The obtained co-occurrence pairs were type- not token-based, i.e. they contained only word-stems.

Disambiguation For each mathematical expression identified as appositional (i.e. preceded by a noun, based on the output of the Stanford parser; see Section 3 for the details on the evaluation set) we considered a local context C consisting of all the nouns appearing in the five word window to the left of a target mathematical expression. For each candidate noun w in the context C we identified the TC terms, *tct*, with the highest semantic similarity according to the similarity metrics described above. In order to identify the TC which best matches the context, we used modified versions of similarity measures presented in [12]. The obtained similarity scores were weighted, summed up, and normalized by the length of the considered context (e.g. the number of nouns found within the five word windows). In weighing the candidates we took into account the distance to the target expression; with the weights decreasing with the distance to the target expression. The similarity was calculated using the following scoring function:

$$Sim(C, TC) = \sum_{w \in C} maxsim(w, TC) \times cw(w), \text{ where}$$

$$maxsim(w, TC) = \max_{tct \in TC} \{sim(w, tct)\} \text{ and } cw(w) \text{ is the weight for the word } w$$

That is, the resulting assigned interpretation is the TC with the highest similarity score between the lexical context and the terms from each of the sets of TC terms.

3 Experiment Setup

We tested the disambiguation method on a manually constructed gold-standard, a set of manually identified and disambiguated appositive mathematical expressions. We compared the algorithm's performance with two baselines which do not use context or have access to limited context information. Below we introduce our evaluation sets, define the performance measures we employed and present the baselines.

Data We conducted an initial evaluation of the approach on all mathematical expressions from one randomly selected document from the arXMLiv containing 451 mathematical expressions [2]; we will refer to this initial evaluation set as *init-set*.

In order to obtain more reliable performance results, in particular, on disambiguation in documents originating from various authors and mathematical sub-areas, we also conducted further evaluation on a set of randomly selected mathematical expressions extracted from a random collection of different documents. This evaluation set (*eval-set*) was constructed as follows: First, we selected 28 random arXMLiv documents which were successfully pre-processed. Second, from each of these documents we selected 20 random mathematical expressions and manually identified the appositive cases among those, obtaining 116 appositive instances. Third, we manually annotated this set with the expected categories, thereby creating a *gold standard*. The gold standard contains 101 disambiguated appositive mathematical expressions. (15 unclear cases from the original set were discarded.)

Performance measures As evaluation metrics we use precision (P), recall (R), $F_{0.5}$, and Mean Reciprocal Rank (MRR). Precision and recall are set-based measures. In classification, precision is the proportion of correctly labelled examples, while recall is the proportion of labelled examples out of all examples. In mathematical expression disambiguation we prefer correct disambiguation over coverage, therefore, we choose $F_{0.5}$ as a combined measure. $F_{0.5}$ is a variant of the harmonic mean of precision and recall which weights precision twice as high as recall. MRR is one of the standard measures used in Information Retrieval for evaluating performance of systems which produce ranked lists of results, for example, ordered lists of documents retrieved in response to a query. It is the inverse of the rank of the expected (best) result item. More specifically,

$$P = \frac{tp}{tp+fp} \qquad R = \frac{tp}{tp+fn} \qquad F_{0.5} = \frac{(0.5^2+1)PR}{0.5^2P+R} \qquad MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

where *tp* are true positive classifications, *fp* are false positives, *fn* are false negatives, and N is the number of evaluated instances.

Baselines We employed two baselines for comparison with our approach. The trivial baseline does not use any context information and simply assigns a random order of categories. The top random category is used in calculating precision. We can consider this as the lower-bound for the performance of the approach. The second baseline uses limited context information: It uses only the noun (*NN*) immediately preceding the target mathematical expression as candidate for disambiguation.

4 Results

Table 3 summarises the results of the evaluation. *Init-set* is our initial set of mathematical expressions from a single document. *Eval-set* is our gold-standard evaluation set. Results are reported for the two baselines: random ranking (*random*) and limited context (*nearest NN*). SW is the approach which uses stop-words to select candidates for similarity computation and *PMI* as the similarity measure. POS-z and POS-*PMI* are the approaches which use POS tags for candidate selection and z and *PMI* as co-occurrence statistics alternative to Dice. δ are the different similarity thresholds.

Init-set was our preliminary evaluation set which served to verify the plausibility of the approach. We did not calculate MRR for this set, however, with the average precision at 81% (for $\delta = 0.9$) we considered the approach promising. With limited access to context information both baselines perform poorly, as expected; recall is not reported because for the baselines some interpretation is always returned. In future experiments we will use

			δ	P	R	$F_{0.5}$	MRR
Init-set	POS-*PMI*		0.6	64.00	31.00	52.77	–
			0.9	81.00	21.00	51.55	–
Eval-set	Baselines	*random*	–	8.51	–	–	0.19
		nearest NN	–	20.21	–	–	0.32
	SW		0.6	53.09	81.13	57.03	0.60
			0.8	61.36	36.49	54.00	0.68
			0.9	63.16	14.28	37.50	0.74
	POS-z		0.0	56.96	44.55	53.96	0.68
			10.0	65.52	37.62	57.06	0.74
			20.0	69.70	22.77	49.36	0.77
	POS-*PMI*		0.6	65.67	43.56	59.62	0.76
			0.8	80.39	40.59	67.21	0.85
			0.9	83.33	39.60	68.26	0.87

Table 3: Evaluation results

other baselines with less limited context information, but limited capabilities of similarity estimation; one plausible baseline could use larger context (e.g. five word window) but calculate only word overlap with Term Clusters, rather than corpus-based similarity.

Considering the limited linguistic preprocessing we employ (the method is based solely on co-occurrence statistics with only stemming, stop-words and largely faulty POS tagging as preprocessing) both the precision results and the ranking results on the evaluation set are encouraging. *PMI* appears to outperform the z-score on this task. Interestingly, the results of the best performing stop-word based model appear comparable not only with the z-score models, but also with the *PMI*-based model at $\delta = 0.6$. This suggests that perhaps further work could be also invested in the knowledge-poor approaches, given the lack of reliable language processing tools for mathematical discourse. Moreover, not surprisingly, with all the measures, the performance is strongly sensitive to the similarity thresholds. We must perform further systematic analyses of the effect of different threshold combinations.

5 Conclusion and Future Work

We can cautiously conclude that the method produces promising results and that, even with limited linguistic information, the lexical context provides useful information in mathematical expression disambiguation. Of course more work and experimentation is needed to further tune the co-occurrence statistics and the similarity metrics. In particular, we are planning to experiment with other corpus-based term association measures. Moreover, we have started to experiment with methods analogous to those presented here, but applied to mathematical expressions which need the *right* context for disambiguation.

Our Term Clusters require further work. In their present state some of the clusters group unrelated terms; see, for instance, the terms grouped under *property*. We are currently working on a more coherent resource with a richer hierarchical structure and with thesaurus-like relations between concepts. With such a resource we can investigate thesaurus-based similarities based on relations such as “broader/narrower concept”. As an initial step we are planning to investigate the relations included in the Cambridge Mathematical Thesaurus.

Similarity could be computed as inversely proportional to the distance between words in the thesaurus hierarchy; short paths between two concepts would indicate high degree of semantic similarity. This is analogous to the way WordNet is used in lexical similarity tasks.

Another resource that we plan to take into account is the "Mathematics Subject Classification" (MSC [1]), a hierarchically organized set of over 5000 mathematical subjects. These could act as Term Cluster Names that cover all of mathematics. We will try to generate the representative terms from the Zentralblatt Math corpus [22], an MSC-classified set of 2.5 million abstracts of mathematical (journal) publications of the last 100 years. We expect to obtain much more accurate disambiguation results from such a resource.

It is clear that linguistic knowledge would help in the disambiguation task. For the left-context appositional cases, the noun phrase part of the nominal group is alone sufficient for disambiguation. However, in order to be able to perform more linguistically-based analysis of the context, we need language processing tools (POS taggers, chunkers, etc.) which are nowadays taken for granted in language processing. Unfortunately, existing tools are typically trained on newspaper text and therefore produce sub-standard results on the mathematical genre. A serious problem here is the lack of annotated data to re-train such tools. We are presently investigating ways of creating a POS-annotated document set and building a specialised POS tagger.

Finally note that our definition of disambiguation still falls significantly short of semantics construction for formulae, where every symbol is interpreted by a semantic concept. Our approach based on the left context and general mathematical resources cannot work since mathematical texts are well-known to introduce notations and concepts as they go along. We would need a deeper discourse analysis that detects notation introductions (and imports for that matter) and brings them to bear locally in the disambiguation process.

Acknowledgements

We would like to thank Deyan Ginev of Jacobs University Bremen without whose many preprocessing scripts it would not have been possible to conduct this study at ease. We would also like to thank three anonymous reviewers for helpful comments.

References and Notes

[1] American Mathematical Society. 2010 Mathematics Subject Classification. `http://www.ams.org/mathscinet/msc/`, 2009.

[2] R. Bañuelos, T. Kulczycki, and P. J. Méndez-Hernández. On the shape of the ground state eigenfunction for stable processes. *Potential Analysis*, 24(2):205–221, 2006. `http://arxiv.org/abs/math/0407260`; Retrieved October 2009.

[3] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th international conference on World Wide Web*, pages 757–766, 2007.

[4] R. Budiu, C. Royer, and P. Pirolli. Modeling information scent: a comparison of LSA, PMI-IR and GLSA similarity measures on common tests and corpora. In *Proceedings of the 8th RIAO Conference*, 2007.

[5] J. Bullinaria and J. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, 2007.

[6] S. Buswell, O. Caprotti, D. P. Carlisle, M. C. Dewar, M. Gaetano, and M. Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004.

[7] The University of Cambridge Mathematical Thesaurus. `http://www.openmath.org`, seen October 2009.

[8] J. Davenport. On writing OpenMath content dictionaries. Technical report, The OpenMath Esprit Project, 2002. `http://www.openmath.org/documents/writingCDs.pdf`.

[9] F.-J. Elmer. Self-organized criticality in the weakly driven Frenkel-Kontorova model. *Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics)*, 56(6):R6225–R6228, 1997. `http://arxiv.org/abs/adap-org/9710002`; Retrieved October 2009.

[10] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* Massachusetts Institute of Technology, Revised version May 1999, 2003.

[11] MathWorld. `http://mathworld.wolfram.com/`, October 2009.

[12] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 775–780. AAAI Press, 2006.

[13] S. Mohammad and G. Hirst. Determining word sense dominance using a thesaurus. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 121–128, 2006.

[14] OpenMath. `http://wiki.openmath.org`, October 2009.

[15] OpenMath Content Dictionaries. `http://wiki.openmath.org/cdnames`, October 2009.

[16] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intellignet Text Processing*, pages 241–257, 2003.

[17] H. Stamerjohanns, M. Kohlhase, D. Ginev, C. David, and B. Miller. Transforming large collections of scientific publications to XML. *Mathematics in Computer Science*, 2009. in press, see `http://kwarc.info/kohlhase/papers/mcs09.pdf`.

[18] Stanford Log-linear Part-Of-Speech Tagger. `http://nlp.stanford.edu/software/tagger.shtml`, October 2009.

[19] M. Strube and S. P. Ponzetto. WikiRelate! computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1419–1424, 2006.

[20] P. D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, pages 491–502, 2001.

[21] WordNet. `http://wordnet.princeton.edu`, October 2009.

[22] Zentralblatt MATH. `http://www.zentralblatt-math.org`, October 2009.