CNGL: Grading Student Answers by Acts of Translation

Ergun Biçici
Centre for Next Generation Localisation,
Dublin City University, Dublin, Ireland.
ebicici@computing.dcu.ie

Josef van Genabith
Centre for Next Generation Localisation,
Dublin City University, Dublin, Ireland.
josef@computing.dcu.ie

Abstract

We invent referential translation machines (RTMs), a computational model for identifying the translation acts between any two data sets with respect to a reference corpus selected in the same domain, which can be used for automatically grading student answers. RTMs make quality and semantic similarity judgments possible by using retrieved relevant training data as interpretants for reaching shared semantics. An MTPP (machine translation performance predictor) model derives features measuring the closeness of the test sentences to the training data, the difficulty of translating them, and the presence of acts of translation involved. We view question answering as translation from the question to the answer, from the question to the reference answer, from the answer to the reference answer, or from the question and the answer to the reference answer. Each view is modeled by an RTM model, giving us a new perspective on the ternary relationship between the question, the answer, and the reference answer. We show that all RTM models contribute and a prediction model based on all four perspectives performs the best. Our prediction model is the 2nd best system on some tasks according to the official results of the Student Response Analysis (SRA 2013) challenge.

1 Automatically Grading Student Answers

We introduce a fully automated student answer grader that performs well in the student response analysis (SRA) task (Dzikovska et al., 2013) and especially well in tasks with unseen answers. Automatic grading can be used for assessing the level of competency for students and estimating the required tutoring effort in e-learning platforms. It can also be used to adapt questions according to the average student performance. Low scored topics can be discussed further in classrooms, enhancing the overall coverage of the course material.

The quality estimation task (QET) (Callison-Burch et al., 2012) aims to develop quality indicators for translations at the sentence-level and predictors without access to the reference. Bicici et al. (2013) develop a top performing machine translation performance predictor (MTPP), which uses machine learning models over features measuring how well the test set matches the training set relying on extrinsic and language independent features.

The student response analysis (SRA) task (Dzikovska et al., 2013) addresses the following problem. Given a question, a known correct reference answer, and a student answer, assess the correctness of the student's answer. The student answers are categorized as correct, partially correct incomplete, contradictory, irrelevant, or non domain, in the 5-way task; as correct, contradictory, or incorrect in the 3-way task; and as correct or incorrect in the 2-way task.

The *student answer correctness prediction problem* involves finding a function f approximating the student answer correctness given the question (Q), the answer (A), and the reference answer (R):

$$f(Q, A, R) \approx q(A, R). \quad (1)$$

We approach f as a supervised learning problem with (Q, A, R, $q(A, R)$) tuples being the training

data and $q(A, R)$ being the target correctness score.

We model the problem as a translation task where one possible interpretation is translating Q (source to translate, S) to R (target translation, T) and evaluating with A (as reference target, RT) (QRA). Since the information appearing in the question may be repeated in the reference answer or may be omitted in the student answer, it also makes sense to concatenate Q and A when translating to R (QARQA). We obtain 4 different perspectives on the ternary relationship between Q, A, and R depending on how we model their relationship as an instance of translation:

$$\begin{aligned} QAR:&\ S = Q, && T = A,\ RT = R. \\ QRA:&\ S = Q, && T = R,\ RT = A. \\ ARA:&\ S = A, && T = R,\ RT = A. \\ QARQA:&\ S = Q + A, && T = R,\ RT = Q + A. \end{aligned}$$

2 The Machine Translation Performance Predictor (MTPP)

In machine translation (MT), pairs of source and target sentences are used for training statistical MT (SMT) models. SMT system performance is affected by the amount of training data used as well as the *closeness* of the test set to the training set. MTPP (Biçici et al., 2013) is a top performing machine translation performance predictor, which uses machine learning models over features measuring how well the test set matches the training set to predict the quality of a translation without using a reference translation. MTPP measures the coverage of individual test sentence features and syntactic structures found in the training set and derives feature functions measuring the closeness of test sentences to the available training data, the difficulty of translating the sentence, and the presence of acts of translation involved.

Features for Translation Acts

MTPP uses n-gram features defined over text or common cover link (CCL) (Seginer, 2007) structures as the basic units of information over which similarity calculations are made. Unsupervised parsing with CCL extracts links from base words to head words, which allow us to obtain structures representing the grammatical information instantiated in the training and test data. Feature functions use statistics involving the training set and the test sentences to determine their closeness. Since they are language independent, MTPP allows quality estimation to be performed extrinsically. Categories for the 283 features used are listed below and their detailed descriptions are presented in (Biçici et al., 2013) where the number of features are given in $\{\#\}$.

- *Coverage* $\{110\}$*:* Measures the degree to which the test features are found in the training set for both S ($\{56\}$) and T ($\{54\}$).
- *Synthetic Translation Performance* $\{6\}$*:* Calculates translation scores achievable according to the n-gram coverage.
- *Length* $\{4\}$*:* Calculates the number of words and characters for S and T and their ratios.
- *Feature Vector Similarity* $\{16\}$*:* Calculates the similarities between vector representations.
- *Perplexity* $\{90\}$*:* Measures the fluency of the sentences according to language models (LM). We use both forward ($\{30\}$) and backward ($\{15\}$) LM based features for S and T.
- *Entropy* $\{4\}$*:* Calculates the distributional similarity of test sentences to the training set.
- *Retrieval Closeness* $\{24\}$*:* Measures the degree to which sentences close to the test set are found in the training set.
- *Diversity* $\{6\}$*:* Measures the diversity of co-occurring features in the training set.
- *IBM1 Translation Probability* $\{16\}$*:* Calculates the translation probability of test sentences using the training set (Brown et al., 1993).
- *Minimum Bayes Retrieval Risk* $\{4\}$*:* Calculates the translation probability for the translation having the minimum Bayes risk among the retrieved training instances.
- *Sentence Translation Performance* $\{3\}$*:* Calculates translation scores obtained according to $q(T, R)$ using BLEU (Papineni et al., 2002), NIST (Doddington, 2002), or F_1 (Biçici and Yuret, 2011b) for q.

3 Referential Translation Machine (RTM)

Referential translation machines (RTMs) we develop provide a computational model for quality and semantic similarity judgments using retrieval of relevant training data (Biçici and Yuret, 2011a; Biçici, 2011) as interpretants for reaching shared semantics (Biçici, 2008). We show that RTM achieves

very good performance in judging the semantic similarity of sentences (Biçici and van Genabith, 2013) and we can also use RTM to automatically assess the correctness of student answers to obtain better results than the baselines proposed by (Dzikovska et al., 2012), which achieve the best performance on some tasks (Dzikovska et al., 2013).

RTM is a computational model for identifying the acts of translation for translating between any given two data sets with respect to a reference corpus selected in the same domain. RTM can be used for automatically grading student answers. An RTM model is based on the selection of common training data relevant and close to both the training set and the test set where the selected relevant set of instances are called the interpretants. Interpretants allow shared semantics to be possible by behaving as a reference point for similarity judgments and providing the context. In semiotics, an interpretant I interprets the signs used to refer to the real objects (Biçici, 2008). RTMs provide a model for computational semantics using interpretants as a reference according to which semantic judgments with translation acts are made. Each RTM model is a data translation model between the instances in the training set and the test set. We use the FDA (Feature Decay Algorithms) instance selection model for selecting the interpretants (Biçici and Yuret, 2011a) from a given corpus, which can be monolingual when modeling paraphrasing acts, in which case the MTPP model is built using the interpretants themselves as both the source and the target side of the parallel corpus. RTMs map the training and test data to a space where translation acts can be identified. We view that acts of translation are ubiquitously used during communication:

Every act of communication is an act of translation (Bliss, 2012).

Translation need not be between different languages and paraphrasing or communication also contain acts of translation. When creating sentences, we use our background knowledge and translate information content according to the current context.

Given a training set $\texttt{train}$, a test set $\texttt{test}$, and some monolingual corpus $\mathcal{C}$, preferably in the same domain as the training and test sets, the RTM steps are:

1. $T = \texttt{train} \cup \texttt{test}$.
2. select$(T, \mathcal{C}) \rightarrow \mathcal{I}$
3. MTPP$(\mathcal{I}, \texttt{train}) \rightarrow \mathcal{F}_{\texttt{train}}$
4. MTPP$(\mathcal{I}, \texttt{test}) \rightarrow \mathcal{F}_{\texttt{test}}$

Step 2 selects the interpretants, $\mathcal{I}$, relevant to the instances in the combined training and test data. Steps 3 and 4 use $\mathcal{I}$ to map $\texttt{train}$ and $\texttt{test}$ to a new space where similarities between the translation acts can be derived more easily. RTM relies on the representativeness of $\mathcal{I}$ as a medium for building translation models for translating between $\texttt{train}$ and $\texttt{test}$.

Our encouraging results in the SRA task provides a greater understanding of the acts of translation we ubiquitously use when communicating and how they can be used to predict the performance of translation, judging the semantic similarity of text, and evaluating the quality of student answers. RTM and MTPP models are not data or language specific and their modeling power and good performance are applicable across different domains and tasks. RTM expands the applicability of MTPP by making it feasible when making monolingual quality and similarity judgments and it enhances the computational scalability by building models over smaller but more relevant training data as interpretants.

4 Experiments

SRA involves the prediction on Beetle (student interactions when learning conceptual knowledge in the basic electricity and electronics domain) and SciEntsBank (science assessment questions) datasets. SciEntsBank is harder due to containing questions from multiple domains (Dzikovska et al., 2012). SRA challenge results are evaluated with the weighted average F_1, $F_1^w = \frac{1}{N}\sum_{c \in C} N_c F_1(c)$ and the macro average F_1, $F_1^m = \frac{1}{|C|}\sum_{c \in C} F_1(c)$ (Dzikovska et al., 2012).

The lexical baseline system is based on measures of lexical overlap using 4 features: the number of overlapping words, F_1, Lesk (Lesk, 1986), and cosine scores over the words when comparing A and R ($\{4\}$) and Q and R ($\{4\}$). Lesk score is calculated as: $L(A, R) = \sum_{p \in M} |p|^2/(|A||R|)$, where M contains the maximal overlapping phrases that match in

A and R and $|p|$ is the length of a phrase [1]. This lexical baseline is highly competitive: no submission performed better in the 2-way Beetle unseen questions task.

4.1 RTM Models

We obtain CNGL results for the SRA task as follows. For each perspective described in Section 1, we build an RTM model. Each RTM model views the SRA task from a different perspective using the 283 features extracted dependent on the interpretants using MTPP. We extract the features both on the training set of 4155 and the test set of 1258 (Q, A, R) sentence triples for the Beetle task and the training set of 5251 and the test set of 5835 (Q, A, R) sentence triples for the SciEntsBank task. The addition of lexical overlap baseline features slightly helps. We use the best reference answer if the reference answer is not identified in the training set.

The training corpus used is the English side of an out-of-domain corpus on European parliamentary discussions, Europarl (Callison-Burch et al., 2012) [2], to which we also add the unique sentences from R. In-domain corpora are likely to improve the performance. We do not perform any linguistic processing or use other external resources. We use only extrinsic features, or features that are ignorant of any information intrinsic to, and dependent on, a given language or domain. We use the training corpus to build a 5-gram target LM. We use ridge regression (RR) and support vector regression (SVR) with RBF kernel (Smola and Schölkopf, 2004). Both of these models learn a regression function using the features to estimate a numerical target value. The parameters that govern the behavior of RR and SVR are the regularization λ for RR and the C, ϵ, and γ parameters for SVR. At testing time, the predictions are bound so as to have scores in the range $[0, 1]$, $[0, 2]$, or $[0, 4]$ and rounded for finding the predicted category.

4.2 Training Results

Table 1 lists the 10-fold cross-validation (CV) results on the training set for RR and SVR for different RTM systems without the parameter optimization for the 5-way task. As we combine different perspectives, the performance improves and we use the QAR+QRA+ARA+QARQA system for our submissions using RR for run 1, SVR for run 2. ARA performs the best among individual perspectives. Each additional perspective adds another 283 features to the representation.

F_1^m / F_1^w	Beetle		SciEntsBank	
Model	RR	SVR	RR	SVR
QAR	.38/.49	.45/.57	.21/.30	.28/.36
QRA	.33/.50	.33/.53	.22/.31	.29/.42
ARA	.45/.54	.50/.60	.21/.30	.30/.38
QARQA	.35/.50	.40/.58	.20/.27	.27/.40
QAR+ARA	.47/.55	.49/.61	.26/.36	.32/.39
QAR+ARA+QARQA	.48/.57	.49/.62	.31/.38	.29/.40
QAR+QRA+ARA+QARQA	.48/.56	.48/.61	.31/.38	.29/.40

Table 1: Performance on the training set without tuning on the 5-way task.

We perform tuning on a subset of the Beetle and SciEntsBank datasets separately after including the baseline lexical overlap features and optimize against the performance evaluated with R^2, the coefficient of determination. SVR performance is given in Table 2. The CNGL system significantly outperforms the lexical overlap baseline in all tasks for Beetle and in the 2-way task for SciEntsBank. For 3-way and 5-way, CNGL performs slightly better.

F_1^m / F_1^w	Beetle			SciEntsBank		
System	2	3	5	2	3	5
Lexical	.74/.75	.53/.56	.46/.53	.61/.64	.43/.55	.29/.41
CNGL	.84/.84	.61/.63	.55/.63	.74/.75	.47/.56	.30/.41

Table 2: Optimized SVR results vs. lexical overlap baseline on the training set for 2-way, 3-way, or 5-way tasks.

4.3 SRA Challenge Results

The SRA task test set also contains instances that belong to unseen questions (uQ) and unseen domains (uD), which make it harder to predict. The training data provided for the task correspond to learning with unseen answers (uA). Table 3 presents the SRA challenge results containing the lexical overlap, our CNGL SVR submission (RR is slightly worse), and the maximum and mean results [3].

According to the official results, CNGL SVR is the 2nd best system based on 5-way evaluation (4th

[1] http://search.cpan.org/dist/Text-Similarity/

[2] We use WMT'13 corpora from www.statmt.org/wmt13/.

[3] Max is not the performance of the best performing system but the maximum result obtained for each metric and subtask.

F_1^m / F_1^w		Beetle		SciEntsBank		
	System	uA	uQ	uA	uQ	uD
2	Lexical	.79/.80	.72/.74	.62/.64	.63/.65	.65/.66
	CNGL	.80/.81	.67/.68	.55/.57	.56/.58	.56/.57
	Mean	.71/.72	.61/.62	.64/.66	.60/.62	.61/.63
	Max	.84/.84	.72/.73	.77/.77	.74/.74	.70/.71
3	Lexical	.55/.58	.48/.50	.40/.52	.39/.52	.42/.55
	CNGL	.57/.59	.45/.47	.33/.38	.31/.37	.31/.36
	Mean	.54/.55	.41/.42	.48/.56	.39/.51	.39/.51
	Max	.72/.73	.58/.60	.65/.71	.47/.63	.49/.62
5	Lexical	.42/.48	.41/.46	.30/.44	.26/.40	.25/.40
	CNGL	.43/.55	.38/.47	.20/.27	.21/.30	.22/.29
	Mean	.44/.51	.34/.40	.34/.46	.24/.38	.26/.37
	Max	.62/.70	.55/.61	.48/.64	.31/.49	.38/.47

Table 3: SRA challenge results: CNGL SVR submission, the lexical overlap baseline, and the maximum and mean results for 2-way, 3-way, or 5-way tasks. uA, uQ, and uD correspond to unseen answers, questions, and domains.

result overall) and the 3rd best system based on 2-way and 3-way evaluation (5th result overall) on the uQ Beetle task. The SVR model performs better than the lexical baseline and the mean result in the Beetle task but performs worse in the SciEntsBank. The lower performance is likely to be due to using an out-of-domain training corpus for building the RTM models and on the uQ and uD tasks, it may also be due to optimizing on the uA task only. The lower performance in SciEntsBank is also due to multiple question domains (Dzikovska et al., 2012).

SVR	Beetle			SciEntsBank		
F_1^w	2	3	5	2	3	5
(a) QAR+ARA	.86	.66	.64	.77	.56	.42
(b) QAR+ARA+QARQA	.86	.66	.65	.77	.57	.45
(c) QAR+QRA+ARA+QARQA	.85	.64	.63	.77	.58	.45
F_1^m	2	3	5	2	3	5
(a) QAR+ARA	.86	.64	.55	.76	.47	.34
(b) QAR+ARA+QARQA	.85	.64	.55	.76	.48	.36
(c) QAR+QRA+ARA+QARQA	.85	.62	.54	.76	.49	.35

Table 4: Improved SVR performance on the training set with tuning for 2-way, 3-way, or 5-way tasks.

4.4 Improved RTM Models

We improve the RTM model with the expansion of our representation by adding the following features:

- *Character n-grams $\{4\}$:* Calculates the cosine between the character n-grams (for n=2,3,4,5) obtained for S and T (Bär et al., 2012).
- *LIX $\{2\}$:* Calculates the LIX readability score (Wikipedia, 2013; Björnsson, 1968) for S and T. [4]

Table 4 lists the improved results on the training set after tuning, which shows about 0.04 increase in all scores when compared with Table 1 and Table 2.

F_1^m/F_1^w		Beetle		SciEntsBank		
	Model	uA	uQ	uA	uQ	uD
2	(a)	.81/.82	.70/.71	.55/.57	.58/.58	.56/.57
	(b)	.80/.81	.71/.72	.69/.70	.54/.56	.56/.58
	(c)	.79/.79	.70/.71	.60/.59	.57/.58	.55/.57
3	(a)	.59/.61	.48/.49	.26/.34	.34/.40	.26/.32
	(b)	.60/.62	.47/.48	.36/.43	.31/.38	.29/.34
	(c)	.58/.60	.46/.48	.41/.48	.30/.39	.29/.34
5	(a)	.47/.56	.37/.45	.19/.22	.22/.33	.22/.29
	(b)	.43/.56	.36/.45	.26/.37	.23/.33	.21/.30
	(c)	.42/.52	.40/.48	.27/.39	.24/.33	.20/.30

Table 5: Improved SVR results on the SRA task test set.

F_1^m/F_1^w		SciEntsBank		
	Model	uA	uQ	uD
2	(a)	.56/.57	.54/.55	.53/.55
	(b)	.57/.58	.53/.54	.56/.57
	(c)	.57/.58	.55/.57	.57/.59
3	(a)	.36/.45	.33/.44	.39/.49
	(b)	.35/.40	.36/.44	.39/.48
	(c)	.37/.46	.36/.48	.40/.50
5	(a)	.24/.34	.23/.33	.26/.39
	(b)	.24/.36	.25/.38	.26/.38
	(c)	.24/.36	.21/.32	.28/.39

Table 6: Improved TREE results on the SRA task test set.

Table 5 presents the improved SVR results on the SRA task test set, which shows about 0.03 increase in all scores when compared with Table 3. SVR becomes the 2nd best system and 2nd best result in 2-way evaluation and the 3rd best system from the top based on 2-way and 3-way evaluation (5th result overall) on the uQ Beetle task.

[4] LIX=$\frac{A}{B} + C\frac{100}{A}$, where A is the number of words, C is words longer than 6 characters, B is words that start or end with any of ".", ":", "!", "?" similar to (Hagström, 2012).

We observe that decision tree regression (Hastie et al., 2009) (TREE) generalizes to uQ and uD domains better than the RR or SVR models especially in the SciEntsBank corpus. Table 6 presents TREE results on the SRA SciEntsBank test set, which shows significant increase in uQ and uD tasks when compared with Table 5.

5 Conclusion

Referential translation machines provide a clean and intuitive computational model for automatically grading student answers by measuring the acts of translation involved and achieve to be the 2nd best system on some tasks in the SRA challenge. RTMs make quality and semantic similarity judgments possible based on the retrieval of relevant training data as interpretants for reaching shared semantics.

Acknowledgments

This work is supported in part by SFI (07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University and in part by the European Commission through the QTLaunchPad FP7 project (No: 296347). We also thank the SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support.

References

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 435–440, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Ergun Biçici and Josef van Genabith. 2013. CNGL-CORE: Referential translation machines for measuring semantic similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*, Atlanta, Georgia, USA, 13-14 June. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011a. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011b. RegMT system for machine translation, system combination, and evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 323–329, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*.

Ergun Biçici. 2011. *The Regression Model of Machine Translation*. Ph.D. thesis, Koç University. Supervisor: Deniz Yuret.

Ergun Biçici. 2008. Consensus ontologies in socially interacting multiagent systems. *Journal of Multiagent and Grid Systems*.

Carl Hugo Björnsson. 1968. *Läsbarhet*. Liber.

Chris Bliss. 2012. Comedy is translation, February. http://www.ted.com/talks/chris_bliss_comedy_is_translation.html.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2012. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, Montréal, Canada, June. Association for Computational Linguistics.

Myroslava O. Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment

challenge. In **SEM 2013: The First Joint Conference on Lexical and Computational Semantics*, Atlanta, Georgia, USA, 13-14 June. Association for Computational Linguistics.

Kenth Hagström. 2012. Swedish readability calculator. https://github.com/keha76/Swedish-Readability-Calculator.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, 2nd edition.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Yoav Seginer. 2007. *Learning Syntactic Structure*. Ph.D. thesis, Universiteit van Amsterdam.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August.

Wikipedia. 2013. Lix. http://en.wikipedia.org/wiki/LIX.