Automatic Event Classification Using Surface Text Features

Hilda Hardy[1], Vika Kanchakouskaya[1], and Tomek Strzalkowski[1,2]

[1]University at Albany, State University of New York
1400 Washington Avenue SS261
Albany, NY 12222
[2]Institute of Computer Science, Polish Academy of Sciences
hhardy@albany.edu, vk813724@albany.edu, tomek@csc.albany.edu

Abstract

Extracting events from documents quickly and accurately is an important goal for many tasks that require language understanding, such as question answering. We present a data-driven method for discovering events and their attributes in a corpus. We further demonstrate that a carefully chosen set of textual features, when used to train some well-known learning algorithms, can approach or exceed the accuracy of hand-crafted patterns for event classification, requiring far less time and expertise. The features can be gathered using lightweight text processing tools. Overall classification accuracy reaches 59.76% for a set of 11 event types.

Introduction

The work presented here is part of a larger research effort focusing on an end-to-end question answering system for intelligence analysts. In analytical question answering, a user poses a complex question such as "What is the history of Iran's nuclear weapons program?" or "Describe the recent conflict between Chechen rebels and the Russian government." In contrast to factoid questions, such as "How tall is the Empire State Building?", a complex analytical question can take a wide variety of syntactic forms.

Whereas answers to factoid questions can be discovered using a finite list of possible answer types, such as *height, person* or *country*, answers to analytical questions are much broader and therefore require other strategies. These may include document retrieval, passage-level clustering, and text framing (Small et al. 2004).

Framing imposes a partial structure on text passages so that the system can compare passages with other passages and with the user's question. **General frames** are created to represent a topic such as *accident, pollution, trade,* etc., captured from the central verb or noun phrase in a passage, together with any number of named entities (LOCATION, PERSON, ORGANIZATION, DATE, etc.) loosely grouped around this topic. **Typed frames** represent specific events such as *Transfer,* with the roles SOURCE, DESTINATION and OBJECT; *Develop,* with AGENT and OBJECT roles; *Attack,* with the roles AGENT, TARGET and INSTRUMENT, etc.

Our system uses event-based, data-driven semantic processing and natural language dialogue, together with an advanced information visualization interface, to deliver accurate answers to analysts' questions, along with related contextual information (Strzalkowski et al. 2005). A foundation of the system is the set of events themselves.

Finding events in text is necessary for applications that require selecting, classifying or filtering such data. The NIST ACE Program (Automatic Content Extraction), for instance, is dedicated to developing technologies that automatically infer meaning from language data (www.nist.gov/speech/tests/ace). Tasks include detecting entities, relations and events. Before events are detected, however, it is necessary to discover or define the categories.

In her influential work on English verb classes, Beth Levin (1993) proposed a set of 48 primary classes, many with subordinate levels, assuming that the syntactic behavior of verbs is semantically determined. Examples of these classes are *Spray/load, Fill, Butter, Remove, Bring and Take (Get, Obtain),* and *Killing (Murder, Poison, Destroy).* Recent corpus annotation efforts, including VerbNet, FrameNet and PropBank, follow Levin's classes to some extent (Fillmore 2001, Kingsbury et al. 2002). These projects aim to provide consistent labeling of verbs and their arguments. Although this research is certainly valuable, our goal is somewhat different: we are interested in finding a smaller, "workable" number of classes representing events of interest in a corpus, that are based not exclusively on verbs but also on important nouns and perhaps adjectives. This number of categories was chosen to cover a particular domain in a representative way that is sufficient to support a QA system. The categories do not capture fine distinctions or rare events, which may be of interest in other applications. Yet we can represent non-typed events in our General frames.

Here we propose a data-driven method for discovering events in a corpus, and we demonstrate how to achieve a high level of classification accuracy using inexpensive textual features for machine learning.

In the following sections we describe our methods for discovering and annotating 11 event types in a corpus, and

how we developed handwritten rules for recognizing some of these events and their roles. We discuss various types of textual features and their relative usefulness for discriminating event classes. Next we show accuracy results for combined feature sets used in training automatic classifiers, comparing these with the scores for the handwritten patterns. Finally, we present accuracy rates for reduced-size data sets.

Event Acquisition from Unstructured Text

We define "event" loosely as an occurrence that changes the world in some way. Events in text can be represented using frames, or structured templates consisting of the event type and its various roles. Types of events are to some extent domain-specific; that is, a corpus of business and financial news can be expected to contain a different set of events than a corpus of medical articles. Yet some events, such as *Transfer* and *Develop*, cross domains easily because of their generic nature.

Our corpus is a set of 37,444 documents, 169 MB, from the Center for Nonproliferation Studies, supplemented by a set of 178,015 documents, 1.5 GB, mined from the web, on the topic of weapons of mass destruction.

We developed a data-driven approach to 1) determine the events of interest in a corpus, as well as the structure those events should take, and 2) extract instances of those events from text.

First we looked at concordances, or contexts, around pairs of named entities, specifically, persons, places and organizations. We used BBN's IdentiFinder to tag the entities (Miller et al. 1999). Contexts consist of content words plus entities as follows: 0-10 words + NE + 1-8 words + NE + 0-10 words. Example sentences are:

Yesterday, 7 December 1941—a date which will live in infamy—the ***United States of America*** *was suddenly and deliberately attacked by naval and air forces of the* ***Empire of Japan***.

Some senior ***Indian*** *ministers had threatened retaliation against* ***Pakistan*** *for its alleged abetment of terrorism in* ***Jammu*** *and* ***Kashmir***.

Considering the most common verbs and sometimes nouns appearing in these contexts, we classified candidate words into event types. With some preliminary event types and roles under consideration, we began annotating randomly selected documents, finalizing the event types and the structures for the frames in the process.

Our four annotators collected a total of 3996 event instances of 11 types, including a *None* category. *None* indicates sentences in which none of the 10 events is present; there may be events of other kinds, or there may be descriptions, opinions, static relations, and so forth.

We list our 10 types of events and their associated roles in Table 1. The role "type" indicates the trigger word or phrase signaling the event.

From the annotated events, we chose at least 100 instances of each of 4 types, *Transfer, Develop, Attack* and *Agree,* to analyze and create syntactic patterns. We designed patterns according to the sequence of syntactic/ semantic elements in the selected passages. Elements such as keywords, lists of weapons, prepositions, noun phrases and named entities were used to create the patterns. For example, the following two tagged sentences each contain an *Attack* event, with the same sequence of key elements:

In the Gaza Strip, <agent>Palestinian gunners</agent> <type>fired</type> <instr>eight mortars</instr> against <target>Jewish settlements</target> overnight <time> Wednesday</time>.

In <time>1988</time> <agent>Saddam</agent> also <type>used</type> <instr>mustard and nerve agents </instr> against <target>Iraqi Kurds</target> at <location> Halabja in northern Iraq</location>.

A pattern that captures both instances is:

pattern: event = "attack", part of speech = "verb", voice = "active"

<NP=Agent> <trigger> + <NP=Instrument> <"against | with | on | at"> + <NP=Target>

Event	***Example Triggers (type)***	***Key Roles***
Agree	treaty, agreement, sign, ratify	PARTIES, INSTR
Assist	helped, supporting, assisted, aid	AGENT, TARGET, INSTR
Attack	attacked, invaded, bombed, destroyed	AGENT, TARGET, INSTR
Develop	construct, develop, manufacture	AGENT, OBJECT
Financial	funded, financed, laundered money	SOURCE, TARGET, QUANTITY
Law-Criminal	arrested, detained, caught, charges	AGENT, TARGET, CHARGE
Law-Nat'l/Int'l	inspectors visited, imposed embargo, passed legislation	AGENT, TARGET, WHAT, CHARGE
Political	election, fired, hired, appointed, voted	AGENT, TARGET, POSITION
Threat	threaten, fear, warned	AGENT, TARGET, INSTR
Transfer	acquire, smuggle, obtain, seize, export	SOURCE, DESTINATION, OBJECT

Table 1: Ten events and selected roles.

Although the hand-designed patterns for extracting events have proven successful, both intrinsically and as a QA-system component that scales well to larger data sets and different domains (see Comparative Results, below), they have limitations. Such a method is labor-intensive and requires a certain level of linguistic expertise. Thus we are implementing a bootstrapping process (based on Strzalkowski and Wang 1996, and Yangarber 2003), which begins with selected high-precision seed rules and uses unsupervised machine learning methods to gather additional rules, in order to increase recall. We are also working on determining the best sets of textual features

and the best supervised learning algorithms for classifying events and their roles. Here we present initial results on feature sets for event classification.

Features for Automatically Classifying Events

Using the annotated event instances above, we collected data on several textual elements that we know to be important in recognizing events and their elements: parts of speech, sentence length, named entities, and patterns of syntactic chunks. We encoded each group of elements into features and fed them into several automatic classifiers. We focused on features that are easy to collect using fast text processing tools, and on well-known, efficient machine-learning algorithms.

Software Resources and Machine Learning Algorithms

We extracted 24 types of named entities from text passages with BBN's IdentiFinder (Miller et al. 1999). We tagged parts of speech using Mark Hepple's tagger from the University of Sheffield (Hepple 2000). We obtained syntactic chunks with the SS Parser, "a fast CFG parser with chunk parsing," from the University of Tokyo (Tsuruoka and Tsujii 2005).

For these experiments we used a collection of machine learning algorithms for data mining tasks implemented by researchers at the University of Waikato, New Zealand, in their Waikato Environment for Knowledge Analysis, or Weka (Witten and Frank 2005).

After experimenting with over two dozen of the algorithms implemented in the Weka toolkit, we selected three of the methods that yielded the best results for our data, plus a baseline method that chose the majority class. Our focus here is not so much on judging the algorithms or their implementations as in finding the most useful features for the classification task. We consider those we have chosen to be good indicators of the performance that can be achieved with selected feature sets on this event classification task.

The first algorithm, *Logistic*, builds a multinomial logistic regression model with a ridge estimator to guard against overfitting by penalizing large coefficients, according to work done by le Cessie and van Houwelingen (1992). The second and third are meta-algorithms. *Vote* combines the results of base classifiers by averaging their probability estimates; we selected four (again, based on performance): a probabilistic Naive Bayes classifier; a REP tree, which builds a decision tree using information gain/variance reduction and prunes it using reduced-error pruning; Random Forest, which constructs a forest of random trees, using a specified number of randomly selected features (Breiman 2001); and Part, which obtains rules from partial decision trees. The third algorithm, *Bagging*, bags a classifier (in this case, a REP tree) to reduce variance (Breiman 1996). All experiments were run with stratified 10-fold cross-validation, unless otherwise indicated.

Word Features (Nouns, Verbs, Adjectives, Pronouns and Prepositions)

Of all the parts of speech, we expected nouns, verbs and adjectives to be the most useful in predicting event types. So from the approximately 4000 annotated instances comprising 11 classes (including *None*), we collected the k most frequent words for each of these 3 parts of speech, minus stopwords, where $20 \leq k \leq 100$. Each word was made into a numeric feature whose value is the number of times the word appears in each instance.

Results for event classification accuracy percentages (Table 2) show that nouns have greater discriminatory power, followed by verbs and adjectives. All were significantly higher than the baseline classifier (majority class). In this and the following tables, numbers shown are the percentages of instances correctly classified.

From this table we can also see a gradual decrease in the rate of accuracy improvement as features are added. Moving from 20 nouns to 40 using the Vote algorithm, for instance, yields an improvement of 7.95 percentage points, whereas moving from 80 to 100 yields an improvement of only 1.63 points. These figures can help dictate the point for a reasonable tradeoff between speed and accuracy.

Algorithm	***Number of Words***				
	100	***80***	***60***	***40***	***20***
	Nouns				
Logistic	53.70	53.05	50.40	48.27	39.01
Vote	***55.06***	53.43	51.15	48.02	40.07
Bagging	53.88	52.43	50.15	47.70	39.89
	Verbs				
Logistic	47.67	47.15	44.02	39.44	36.11
Vote	***48.62***	47.87	44.27	39.21	36.29
Bagging	48.05	47.27	43.87	39.19	36.09
	Adjectives				
Logistic	31.93	31.06	30.93	30.0	30.38
Vote	***33.21***	32.66	32.08	30.83	30.46
Bagging	32.51	31.91	31.88	30.96	30.53

Table 2: Classification accuracy percentages, k most frequent nouns, verbs and adjectives (baseline: 21.45%).

For convenience we grouped together pronouns and prepositions, counting the number of pronouns found in each instance, as well as the numbers of each of 14 prepositions. These also proved to have discriminatory value (Logistic: 34.91%; Vote: 35.69%; Bagging: 35.99%; Baseline: 21.45%).

Sentence Length and Named Entity Features

We computed the number of words in a sentence and tested this feature alone for the set of event instances (11 classes, 3996 instances). For the few cases in which one or more

elements fell outside the sentence boundaries, we chose the sentence containing the majority of the event elements.

Table 3 shows classification accuracy rates for sentence length. Although the numbers are only slightly above baseline, the percentages are significant for a single feature. Indeed, individual feature sets become more powerful when used together to train automatic classifiers.

Knowing the importance of named entities as slot-fillers in event frames, we conducted experiments to investigate how useful NEs are in distinguishing event classes. The NE tagger IdentiFinder labels 24 types of entities in text, including *Geo-Political Entity, Location, Person, Time*, etc. As with the word features, each entity became a numeric feature. We eliminated the 6 least frequent types of entities in our data without significantly affecting accuracy rates. With 18 entities using the *Vote* algorithm, accuracy was 39.94% (baseline 21.45%).

Algorithm	***Accuracy***
Logistic	28.53%
Vote	***29.33***
Bagging	29.08
Baseline (majority class)	21.45

Table 3: Classification accuracy, sentence length.

Syntactic Chunk Patterns

To determine the relative importance of syntactic patterns for event classification, we labeled each event instance with POS tags, then built parse trees using a fast chunk parser (Tsuruoka and Tsujii 2005). We chose 6 types of phrases to use in our patterns: *NP, VP, Verb, PP, ADVP* and *Other*. *Verb* and *Other* represent sequences of one or more tokens, including words and punctuation, that the parser did not attach to a chunk. Beginning with each clause identified in the sentence, we collected the k most frequent sequences of phrases with length 2 through 5, where $10 \leq k \leq 50$ (no overlaps). To establish a cutoff point for long sentences with many nested phrases, we descended into each parse tree, stripping the outer layers, until each phrase consisted of 8 or fewer words.

Examples of the patterns are *NP-VP, NP-Verb-NP-PP, NP-ADVP-VP, Other-NP-VP,* etc. As with the features described above, we recorded the number of occurrences in each event instance.

Accuracy rates are not as high as we had hoped to see for this set of features—the best performance is 25.83%, only 4.38 percentage points above baseline for the set of 11 event classes when tested with the 50 most frequent patterns. Nor did we see significant differences when moving from the largest number of patterns (50) to the smallest (10). Perhaps these results reflect errors in the POS tagger or the parser or both. We think further investigation of syntactic patterns is worthwhile for this area of research, because there is a well-known connection between the meaning of a sentence and its syntactic form.

Results for Combined Feature Sets

Next, we combined the textual features in various ways, to evaluate accuracy rates for all the features as well as for reduced feature sets. We continued to use the 3 machine-learning algorithms described above, plus the majority-class baseline. Features for syntactic patterns were not included in these tests.

Table 4 shows our feature combinations. The results for the individual feature sets, reported in previous sections, served as a guide for constructing the combined feature sets.

N	***Vb***	***Adj***	***P&P***	***SL***	***NE***	***Total features***
50	40	40	15	1	24	**170**
50	20	20	15	1	18	**124**
40	20	20	15	1	18	**114**
30	20	10	15	1	18	**94**

Table 4: Feature sets used in combined tests. *N* = number of noun features, *Vb* = verbs, *Adj* = adjectives, *P&P* = pronouns and prepositions, *SL* = sentence length, *NE* = named entities.

The highest accuracy was achieved with the *Logistic Regression* algorithm and a set of 124 features (Table 5). The *Vote* algorithm performed nearly as well, however, and when we reduced the number of features for both algorithms, the drop in accuracy was slight. Often a reduced feature set is beneficial in terms of time savings, with only a slight reduction in accuracy.

Algorithm	***Number of Features***			
	170	***124***	***114***	***94***
Logistic	59.13%	***59.76%***	59.46%	58.61%
Vote	58.98	58.81	58.56	56.58
Bagging	52.93	52.93	52.98	52.33
Baseline	21.45	21.45	21.45	21.45

Table 5: Classification accuracy, combined feature sets.

A closer look at the top results, where we saw an overall rate of 59.76% correctly classified instances, shows Precision rates above 60% for 6 of the 11 categories, and F-measure scores above 60% for the event classes *Agree, Transfer, Attack*, and the *None* category (Table 6). Here we also show the number of instances for each event. Not surprisingly, the poorly represented classes yielded lower classification accuracy rates.

After we removed these under-represented event classes, we saw Precision rates between 64.3% and 75.5%, and F-Measure scores between 54.2% and 76.2% (Table 7).

A confusion matrix for the same experiment (Table 8) shows that misclassifications occur for events that have similar elements. *Transfer* and *Develop*, for example, might refer to the same types of items. A country might transfer an item it has first manufactured. Weapons used in

Attacks might in other cases be transferred by the same entity. A *Threat* is often seen as an *Attack* not carried out. And the *None* category includes a variety of examples that together touch on all the event topics.

Class	No.	Precision	Recall	F-Measure
NONE	827	0.632	0.811	0.711
TRN	857	0.607	0.614	0.61
DEV	540	0.614	0.57	0.591
ATT	687	0.613	0.64	0.626
FIN	143	0.51	0.371	0.429
THR	177	0.61	0.469	0.53
AST	210	0.326	0.21	0.255
POL	46	0.216	0.174	0.193
LEI	134	0.262	0.164	0.202
LEC	50	0.222	0.16	0.186
AGR	325	0.712	0.692	0.702

Table 6: Detailed accuracy by class, *Logistic* algorithm, 124 features (*No.* indicates number of instances), 11 event classes.

Class	No.	Precision	Recall	F-Measure
NONE	827	0.698	0.839	0.762
TRN	857	0.694	0.684	0.689
DEV	540	0.682	0.593	0.634
ATT	687	0.717	0.705	0.711
THR	177	0.643	0.469	0.542
AGR	325	0.755	0.702	0.727

Table 7: Detailed accuracy by class, *Logistic* algorithm, 124 features (*No.* indicates number of instances), 6 event classes.

	Classified as:					
	NONE	TRN	DEV	ATT	THR	AGR
NONE	694	34	22	53	6	18
TRN	92	586	85	65	8	21
DEV	64	112	320	18	4	22
ATT	99	59	13	484	24	8
THR	18	26	5	40	83	5
AGR	27	27	24	15	4	228

Table 8: Confusion matrix for *Logistic* algorithm, 124 features, 6 classes.

Comparative Results, Manual vs. Automatic

For comparison we present evaluation results for the hand-crafted rules described above, with one caveat. The "Manual" F-scores for each event type shown below indicate an overall average of the separate scores for each role in each event. Therefore, they cannot be directly matched with the "Auto" results, which indicate only the event classification.

Once we implement ML techniques to identify the roles for each event instance, we expect ML ("Auto") scores to decline slightly. Nevertheless, we believe that the comparison is valid here as an indication of the relative performance of the two approaches. We maintain that automatic classification can approach or exceed the accuracy of hand-written rules, at far less expense.

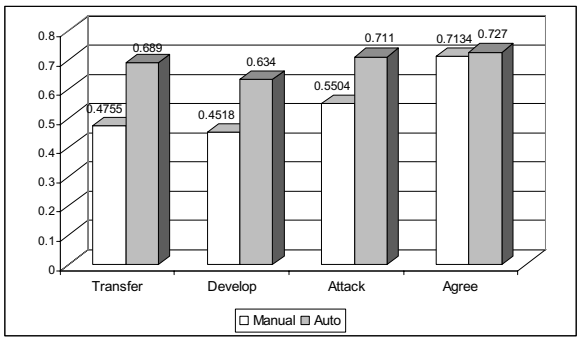

Figure 1: Comparison of F-measure scores, manual vs. automatic techniques.

Reducing the Size of the Data Set

Finally, we ran a series of experiments to see how the size of the data set affects classification accuracy. As mentioned above, we collected a total of 3996 annotated instances of 11 event types. We have already seen in Table 6 that low-frequency classes have correspondingly low precision and recall scores. But given the fact that manual annotation is expensive in terms of time spent training annotators, performing the annotation, maintaining consistency among annotators, and cleaning the data, we would like to be able to annotate less, or supplement manual annotation with automated methods, or both.

We adjusted the split of training/testing instances in 10% increments and ran tests with three machine-learning algorithms, *Logistic Regression, Bagging* and *Vote*, to see what would happen to the overall classification accuracy. Figure 2 shows accuracy rates for the various split percentages. The numbers indicate the percentage of the data used for training. (The remainder of the data was used for testing.)

The results for 90% of the training data were achieved using 10-fold cross-validation (as in all the previous experiments), and the figures for 100% of the training data show results when both training and testing were done on the full data set. These numbers indicate the best performance possible for the selected features on the data.

From these figures we can see that using 60% of the data set is nearly as good as using 90% in terms of overall classification accuracy. We expect that the accuracy of the low-frequency classes could be improved by gathering more instances of those events, perhaps by using a targeted search within the corpus, to bring the accuracy numbers in line with those of the more frequent events.

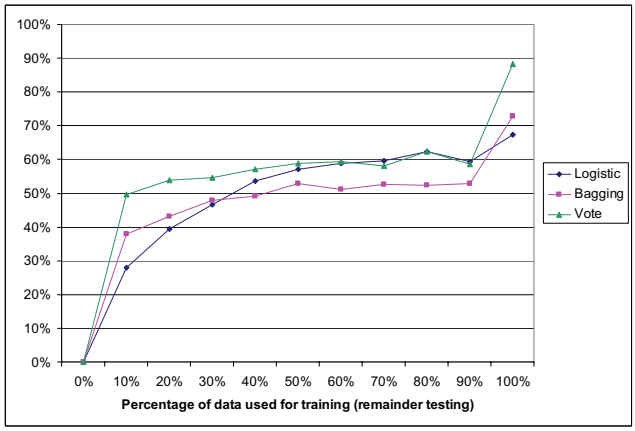

Figure 2: Classification accuracy according to size of training set (114 features).

Future Work

We have shown that a careful selection of textual features for event classification can yield accuracy rates similar to those achieved using hand-crafted patterns, with less expense. Nouns, prepositions and named entities are particularly useful discriminatory features. We hope to revisit the syntactic pattern features with other fast parsing resources, both as an aid to automatic event classification, and to help classify the attributes or roles in each event.

A larger goal is to automate the event acquisition process, so that we can move to an unknown corpus, discover a set of important topical events, and extract instances from text. We hope to leverage the data we already have, especially for the more generic events, to discover similar event classes in new corpora. We also intend to work from the elements we know to be important in distinguishing events in our current corpus, to discover unknown events in other domains.

Acknowledgments

This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program. The authors wish to thank Ralph Weischedel for the use of IdentiFinder; and Sharon Small, Nobuyuki Shimizu, Tracy Janack, Rob Salkin, Sean Ryan, Ashkhen Pogosyan and Ben Carle for their help.

References

Breiman, L. 2001. Random Forests. *Machine Learning* 45(1):5-32.

Breiman, L. 1996. Bagging Predictors. *Machine Learning* 24(2):123-140.

Fillmore, C., and Baker, C.F. 2001. Frame Semantics for Text Understanding. *WordNet Workshop at NAACL.*

Hepple, M. 2000. Independence and Commitment: Assumptions for Rapid Training and Execution of Rule-based Part-of-Speech Taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000),* 278-285. Hong Kong.

Kingsbury, P.; Palmer, M.; and Marcus, M. 2002. Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference (HLT '02).*

le Cessie, S., and van Houwelingen, J.C. 1992. Ridge Estimators in Logistic Regression. *Applied Statistics* 41(1):191-201.

Levin, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, IL.

Miller, D.; Schwartz, R.; Weischedel, R.; and Stone, R. 1999. Named Entity Extraction from Broadcast News. In *Proceedings of DARPA Broadcast News Workshop,* Herndon, VA.

Small, S.; Strzalkowski, T.; Liu, T.; Ryan, S.; Salkin, R.; Shimizu, N.; Kantor, P.; Kelly, D.; and Wacholder, N. 2004. HITIQA: Towards Analytical Question Answering. In *Proceedings of the 20th International Conference on Computational Linguistics, Coling 2004,* Geneva, Switzerland.

Strzalkowski, T.; Small, S.; Hardy, H.; Yamrom, B.; Liu, T.; Kantor, P.; Ng, K.B.; and Wacholder, N. 2005. HITIQA: A Question Answering Analytical Tool. In *Proceedings of the International Conference on Intelligence Analysis,* McLean, VA.

Strzalkowski, T., and Wang, J. 1996. A Self-Learning Universal Concept Spotter. In *Proceedings of Coling 1996*.

Tsuruoka, Y., and Tsujii, J. 2005. Chunk Parsing Revisited**.** In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005),* 133-140.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques,* 2nd Edition. San Francisco, CA: Morgan Kaufmann.

Yangarber, R. 2003. Counter-Training in Discovery of Semantic Patterns. In *Proceedings of ACL-2003.* Sapporo, Japan.