

# Classifying Action Items for Semantic Email

Simon Scerri<sup>1</sup>, Gerhard Gossen<sup>1</sup>, Brian Davis<sup>1</sup>, Siegfried Handschuh<sup>1</sup>

<sup>1</sup> Digital Enterprise Research Institute,  
National University of Ireland Galway  
IDA Business Park, Galway, Ireland.

E-mail: simon.scerri@deri.org, gerhard.gossen@deri.org, brian.davis@deri.org, siegfried.handschuh@deri.org

## Abstract

Email can be considered as a virtual working environment in which users are constantly struggling to manage the vast amount of exchanged data. Although most of this data belongs to well-defined workflows, these are implicit and largely unsupported by existing email clients. Semanta provides this support by enabling *Semantic Email* – email enhanced with machine-processable metadata about specific types of email *Action Items* (e.g. *Task Assignment*, *Meeting Proposal*). In the larger picture, these items form part of ad-hoc workflows (e.g. *Task Delegation*, *Meeting Scheduling*). Semanta is faced with a knowledge-acquisition bottleneck, as users cannot be expected to annotate each action item, and their automatic recognition proves difficult. This paper focuses on applying computationally treatable aspects of speech act theory for the classification of email action items. A rule-based classification model is employed, based on the presence or form of a number of linguistic features. The technology's evaluation suggests that whereas full automation is not feasible, the results are good enough to be presented as suggestions for the user to review. In addition the rule-based system will bootstrap a machine learning system that is currently in development, to generate the initial training sets which are then improved through the user's reviewing.

## 1. Introduction

A lot of work in today's business environments depends on online communication. Tasks are created, managed and delegated; meetings requested and scheduled; important data exchanged – all via online communication media and on a daily basis. Communication media like email and instant messaging (IM) have become essential virtual working environments wherein people manage large amounts of data within a multitude of implicit workflows. Keeping track of these workflows is not easy, and frequently people become inundated with too much data than they can possibly handle – a problem termed as information overload (Whittaker & Sidner, 1996). As a result, questions get ignored, commitments forgotten and in general, collaboration and ultimately productivity suffer. There have been numerous attempts at automatically extracting action items, to-do's and general commitments from text pertaining to electronic conversations, especially with regards to ones taking place over email. In particular we have worked on models (Scerri & Handschuh & Decker, 2008) that conceptualise these items and outline their expected workflows. These models were implemented within Semanta – extensions to popular email clients which strive to support the user with the management of email workflows (Scerri et al., 2009). However, from a practical point of view Semanta cannot rely on the end-user to recognize, classify and annotate each single email action item. Therefore at the least, partial automation is required. In this paper we introduce a rule-based classification model that classifies email segments into a predetermined set of action items, and discuss the results of its evaluation.

## 2. Background

The set of action items used for the classification of email text<sup>1</sup> are instances of the *Speech Act Model* provided in the *sMail Framework* – a conceptual framework for semantic email, and presented in earlier work (Scerri et al., 2008). This model is based on aspects of the Speech Act Theory (Searle, 1969), which states that every utterance implies an action by the speaker with varying effects on both the speaker and the hearer. When applied to electronic conversations, the sender and the recipient perform the roles of the speaker/hearer whereas textual phrases function as utterances. Action items in the model consist of three parameters:

- *Action* – what is being performed e.g. a request, a notification or an assignment
- *Object* – the object of the action e.g. a request for a meeting
- *Subject* – the subject/agent of the object if applicable e.g. who will/would attend the meeting

Actions consist of *Request* – an action requiring a reply from the recipient (e.g. a question); *Assign* – an action requiring an activity but no reply (e.g. an order or a commitment); *Suggest* – an action involving an optional activity; and *Deliver* – the action of delivering data. Objects are categorised into *Activities* (*Task* and *Event*) and *Data* (*Information* and *Resource*). The subject parameter is only applicable to activities (being the task performer(s) or the event participant(s) – i.e. *Sender*, *Recipient*, *Both*). Thus, a request for permission to attend

---

<sup>1</sup> Limited to email messages in the English language

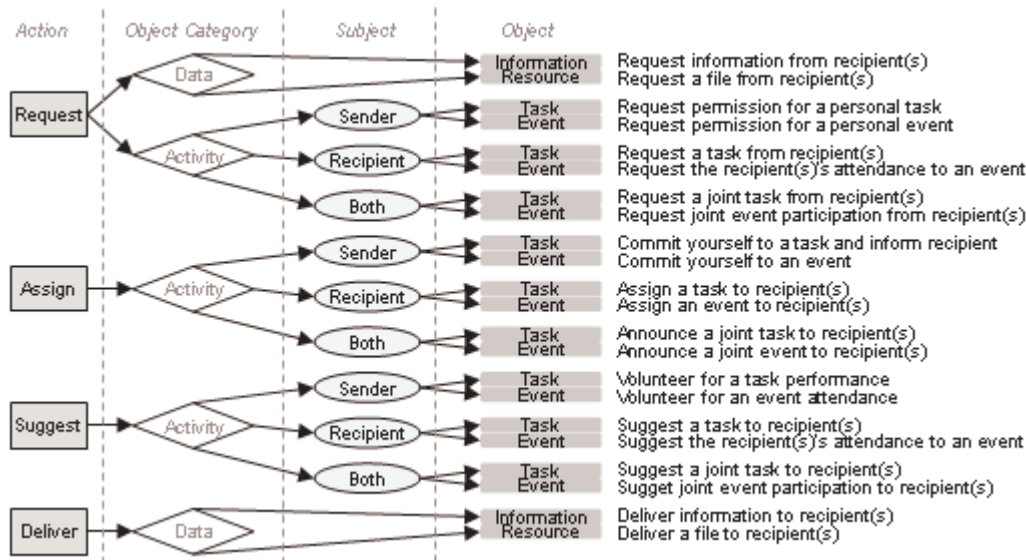


Figure 1: The 22 action item instances for the classification task, with a short description

an event is represented as a (Request, Event, Sender), an order to perform a joint task as an (Assign, Task, Both), and a request for information can be represented as a (Request, Information, Ø). The basic 22 combinations of these parameters (i.e. the email action items) together with a brief description are shown in Fig. 1.

When applied in practice the model can generate more combinations, as in reality the subject will be a member of the power set of the sender and all recipients e.g. (Request, Event, {Sender, Recipient<sub>1</sub>, Recipient<sub>2</sub>,..., Recipient<sub>n</sub>}) for a joint event request involving the sender and two other contacts. The model in question was evaluated in an experiment which measured the inter-annotator agreement between human annotators (Scerri et al., 2008). The annotators had the task of identifying instances of the 22 action items within email from an email corpus. The resulting agreement rate, at 0.811, proved that our model was more appropriate for the task in hand than earlier speech act theory-inspired models, such as Carvalho & Cohen (2006). However, the result indicates the difficulty of the classification task, even when performed by humans.

### 3. Classification Model

The implemented technology for the automatic classification of email action items is based on a classification model that considers the following 5 linguistic, grammatical and syntactical features:

**Modality** – Sentence modality deals with different phrase types, of which we consider the following three: *Declaratives*, *Imperatives* and *Interrogatives*. Whereas most interrogative sentences/clauses are easily recognised by the presence of a question mark at the end, to differ between the two remaining types our classification model considers modal verbs (e.g. must,

will), especially those expressing the concepts of *Possibility* and *Necessity*, which are roughly equal to our ‘Suggest’ and ‘Assign’ actions respectively.

**Verb Category** – Verbs are used to express an *Action*, an *Occurrence* or a *State of being*. Since we attempt to recognise action items, our main interest is action verbs. Our model differs between the following two categories of action verbs, which we refer to as *Activity Verbs* - representing events and tasks (e.g. go, prepare); and *Communicative Verbs* - implying actions specific to electronic communication (e.g. send, forward, attach).

**Grammatical Tense** – The tense morpheme specifies the time at/during which the descriptive content of the sentence in question holds (Ogihara, 2007). There are different opinions when it comes to categorizing tenses in the English language (Comrie, 1985). We adhere to the two-tense approach – *Past* and *non-Past*, as we are mostly interested in actions that occur in the non-past.

**Negation** – From a pragmatic point of view, negation usually expresses the exact opposite of what otherwise the statement would convey, i.e. impossibility instead of possibility, prohibition instead of permissibility (Moeschler, 1992). Both nouns and verbs can be negated via the use of a negative adjective, a negative pronoun or a negative adverb.

**Semantic Role** – When dealing with action verbs we are interested in the subject of that action. We are concerned with the semantic rather than grammatical roles, i.e. the *Agent* and the *Patient*. The grammatical person for both roles - *First*, *First Plural*, *Second* or *Third Person* has also a bearing on the classification task.

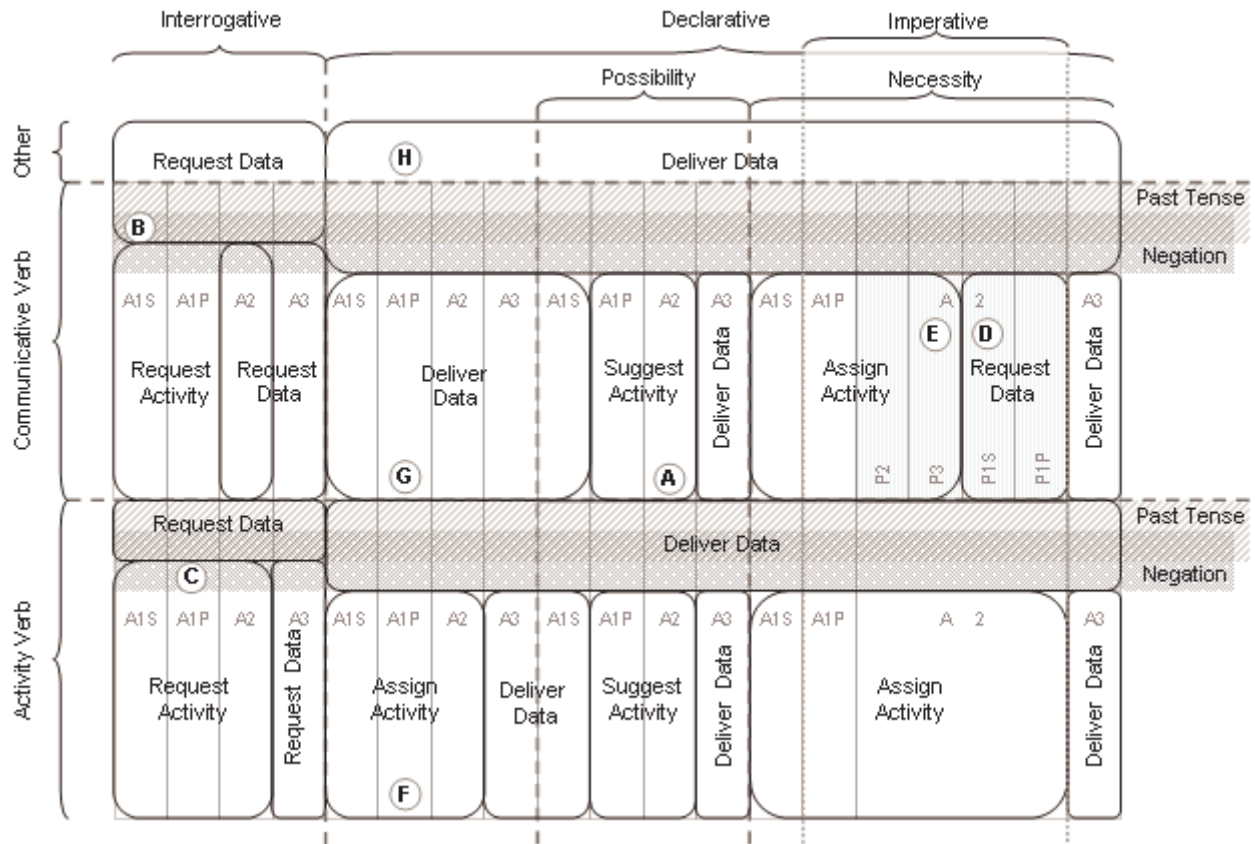


Figure 2: Graphical representation of the simplified classification model, classifying clauses into five class categories

The model as depicted in Fig. 2. is only meant to provide an insight of how all the previous features can all be factored in to attempt a decision on what action item best represents a piece of text. The illustration breaks down the linguistic space into a number of dimensions, given the presence/form of the above features. Grammatical clauses are classified into exactly one of the resulting classification classes (action items). For simplification purposes, we do not include all 22 classes in the figure, but instead abstract over five broad categories which disregard the different object types and the subject parameter entirely. The resulting five categories are thus equivalent to the five pairs of ‘action - object category’ in Fig. 1, i.e. *Request Data*, *Request Activity*, *Suggest Activity*, *Assign Activity* and *Deliver Data*.

Modality splits the space vertically into interrogatives and declaratives. We further differentiate between declarative statements having a possibility modal, those having a necessity modal (which subsume imperative statements) and the rest. The space is split horizontally between communicative verbs, activity verbs and their complement. Statements having activity or communicative verbs are segmented given the agent semantic role where *A1S* stands for Agent 1st Person Singular; *A1P* for Agent 1st Person Plural; and *A2*, *A3* for Agent 2nd and 3rd Person respectively. In order to keep the figure as simple as possible the patient role (similarly *P1S*, *P1P*, *P2*, *P3*) only features in one quadrant, in order to provide adequate examples in the

next section. Negation and Grammatical Tense (past tense) are represented as (overlapping) horizontal shades of grey across statements with communicative and activity verbs.

In the following two subsections we will first demonstrate how the classification model can in theory be employed to classify textual clauses into a specific action item, followed by a description of its practical implementation.

### 3.1 Classification Examples

Table 1 demonstrates how the classification model can be employed via eight practical examples. The examples are matched against one of the implemented pattern rules (in BNF style<sup>2</sup>) as introduced in the next subsection. The bold tokens in example A are matched against the LHS of the rule below to classify as a task suggestion action item (Suggest, Task, Recipient); were “You” is recognised as a second person agent, “should” as a possibility modal, and “forward” as an (electronic) communicative verb. The classification can be mapped to Fig. 2 by focusing on the intersection between the horizontal communicative verb segment and the vertical declarative/possibility modal segment. The presence of a second person agent in the text places it in its shown position in Fig. 2-Ex. A. As the action item categories in the figure disregard the subject parameter, the statement

<sup>2</sup> <http://foldoc.org/?Backus-Naur+Form>

Table 1: Examples of text as classified by rules based on the classification model

A	“You should forward it to me.” ([A1]   [A2] [PosMod] ([TaskV]   [EventV]   [CommV]) : Suggest Task to Recipient
B	“Hadn’t I sent you the file?” ([PastAux]   [PastAuxNegation]) ([A1S]   [A1P]   [A2]   [A3]) [CommV] ({Person})? [Q] : Request Information
C	“Can we not meet to plan ahead today?” ([Modal]   ([ModalNegation]))? ([A1S]   [A1P]   [A2]) [Negation]? ([TaskV]   [EventV]) [Q] : Request Joint Event
D	“you still have to send me the document!” [A2] ([NecMod])? [CommV] ([P1S]   P1P)? : Request Resource
E	“You must email them the document.” [A2] [NecMod] [CommV] ([P2]   [P3]   [Entity]) : Assign Task to Recipient
F	“We are going to attend the meeting,” ([A1S]   [A1P]   [A2]) [NecMod]? ([TaskV]   [EventV]) : Assign Joint Event
G	“We are sending you the files...” ([A1S]   [A1P]) [NecMod]? [CommV] [P2]? : Deliver Resource
H	“We are happy.” Catcher rule – all unclassified declarative clauses : Deliver Information

is shown as a *Suggest Activity*.

The question mark at the end of example B indicates an interrogative statement. The “sent” verb being of type communicative, preceded by a first person agent would have placed the sentence in the *Request Activity* quadrant in Fig. 2-B. However the question's past tense, identified by the verb and the past auxiliary “Hadn't” reduces it to a simple information request action item (Request, Information, Ø) within the *Request Data* category. Although the past auxiliary is negated, negation alone would not have effected the statement. Thus “Won't I send you the file?” or “Will I not send you the file” would still have classified as a personal task request action item.

Example C is similar to B, with the difference that the verb “meet” is recognised as a non-past activity verb. Being an interrogative, this places the statement in the lower horizontal segment. Although “Can” is recognised as a modal verb, these verbs only effect non-interrogative statement. Once again, negation does not effect the classification in this case, and the presence of the first person plural agent “we” results in a joint event request action item (Request, Event, Both) shown as a *Request Activity* in Fig.2-C.

Examples D and E are both declarative statements with a necessity modal, and they differ only with respect to the patient role. Whereas D is a request for the recipient (second person agent - “you”) to perform a communicative action verb (“send”) to the sender (first person singular patient - “me”), E is a request for the recipient to perform a communicative action (“email”) to a third party (third person patient - “them”). Whereas D is classified as a resource request (Request, Resource, Ø), E is classified as a task assignment action item (Assign, Task, Recipient). The resulting difference in classification is also illustrated in Fig. 2, mapped to the

broader categories of *Request Data* (Fig.2-D) and *Assign Activity* (Fig.2-E).

Example F and G differ mostly due to the verb type. The presence of a first person plural agent (“We”) followed by a non-past activity verb (“attend”) classifies as a joint event assignment (Assign, Event, Both) as shown by Fig.2-F. Instead, G includes a non-past communicative verb (“sending”) followed by a second person patient (“you”). This classifies the statement as a resource delivery action item (Deliver, Resource, Ø), shown in Fig.2-G as an instance of the *Deliver Data* category.

All non-interrogative clauses which remain unclassified by the implemented pattern rules, e.g. example H, are classified as an information delivery (Deliver, Information, Ø), also shown as a *Deliver Data* in Fig.2-H. However, as any non-action item would be classified as an information delivery, instances of this class are not returned to the user for review.

### 3.2 Implementation

The classification model has been implemented as a rule-based classifier in GATE (Cunningham et al., 2002). The classifier consists of an ANNIE Corpus IE Pipeline. The pipeline (Fig. 3) consists of a:

- 1) Standard GATE English Tokeniser
- 2) Standard Sentence Splitter
- 3) Standard Hepple POS Tagger
- 4) ANNIE Gazetteer Lookup
- 5) Standard Named Entity Transducer
- 6) JAPE Speech Act Grammars Set



After the text is tokenised and split into sentences, it is forwarded to the POS tagger, which assigns a part of

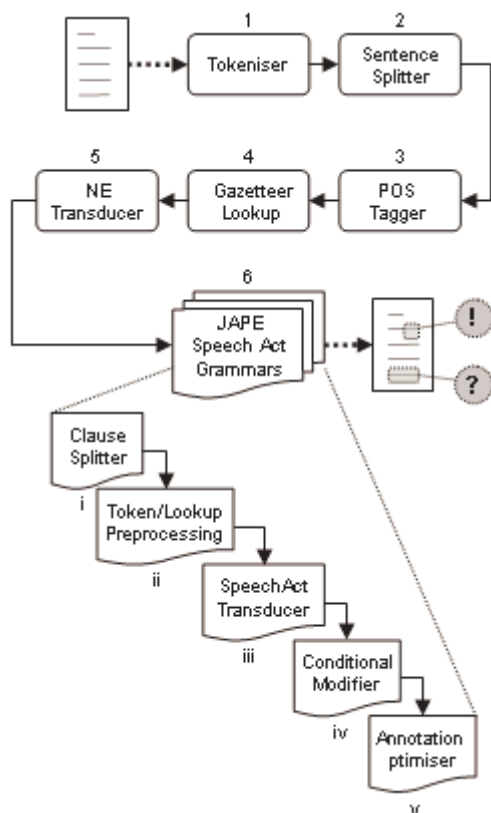


Figure 3: The Speech Act Classification Pipeline

speech category to each token. In particular, the POS tagger recognises past tense verb inflections. The gazetteer lookup performs customised finite state lookup for key-phrases, including trigger words/phrases for linguistic features such as negation, modal verbs, grammatical person and our two different verb categories. The Named Entity Transducer performs named entity identification and is particularly useful in the recognition of person references in the text.

The set of hand-coded Java Annotation Patterns Engine (JAPE) (Cunningham & Maynard & Tablan, 2000) grammars is the most important component in the pipeline. The grammars provide pattern rules such as the ones provided for the examples in Table 1. The rules match combinations of the linguistic/semantic annotations output by the previous components in the pipeline, to classify clauses into one of our action items. The grammars themselves constitute a cascade of finite state transducers over patterns of annotations, such that the output of one transducer becomes the input of the next, as follows:

- i. The *Clause Splitter* is a personal modification of the sentence splitter that splits sentences into individual clauses on which we perform classification.
- ii. The *Clause Splitter* is a custom modification of the

sentence splitter that splits sentences into individual clauses – upon which we want to perform classification our classification.

- iii. *Token and Lookup Preprocessor* binds special kinds of tokens/gazetteer entries to intermediate annotations (e.g. groups modal verbs, grammatical persons by category)
- iv. The *Speech Act Transducer* matches combinations of intermediate annotations to a one of the model’s speech act class. This is where most of the pattern matching is performed.
- v. The *Conditional Modifier* changes some of the identified action items based on preceding/succeeding conditional modifiers, e.g. the presence of an ‘if-then’ clause before an identified task assignment, changes the classification to a task suggestion or a task request, depending on the context.
- vi. The *Annotation Optimiser* extends action items to cover whole sentences, and to group together consecutive identical items.

Each JAPE transducer consists of a collection of phases which in turn contain pattern/action rules. The left hand side (LHS) of the rule is written in BNF style, similar to the examples in Table 1, whereas the right hand side (RHS) consists of annotation-binding variables within a block of JAVA code, which can subsequently be manipulated as desired. JAPE rules can fire in various ways depending on the desired behavior e.g., based on textual ordering, priority or longest match. The Speech Act Transducer alone consists of 58 rules within 14 different phases, such that text matched in the initial phases may not be considered later.

#### 4. Evaluation

To evaluate our classification rules, we wanted to compare automatically- to manually-generated annotations. We employed twelve people to review automatic annotations generated for at least 8 email messages. The evaluators were introduced to the available classification classes (Fig. 1) prior to the task. The evaluation page, with a link to the active web service is available online<sup>3</sup>. The reviewing consisted of rating the classified action items, and annotating the missing action items manually. Each action item returned could be rated using a 4-point Likert scale, two for correct annotations and two for false positives. The reason for multiple positive and negative ratings is that the classes for the classification task are not always

<sup>3</sup> <http://smile.deri.ie/projects/smali/Evaluation>



Figure 4: Main results of the evaluation

outright good or bad, and there are cases where multiple action items can apply to a clause or phrase, with different levels of suitability. Additionally, we believe the author of the email is in the best position to determine whether the classified action items apply, and to which degree. For example, although a user might be happiest with a *Request Event* action item for “Can you discuss this with them?”, they might also think that a *Request Task* is also suitable, and that a *Request Information* is still relevant although to a lesser degree.

The evaluators ran the classification rules over a total of 116 emails, and rated 194 classified action items. A further 74 items were manually annotated by the user. Positive ratings, representing correctly classified action items, amounted to 41% (Fig.4). Negative ratings, representing false positives, amounted to 31%. Missing action items amounted to 28%. We obtained an *F*-measure, weighing precision (0.56) and recall (0.60) equally, of 0.58. This result needs to be interpreted in the light of the rate obtained for the inter-annotator agreement experiment referenced earlier, which at 08.11 indicated the difficulty of the classification task, even when performed by humans. Therefore an *f*-measure of 58% means that although the automatic classification fares very well, it is still not reliable enough to be used for automatic annotation. This has been taken into account in the current implementation of Semanta, which only promises semi-automatic email annotation by providing the results as suggestions to the user.

## 5. Conclusions and Future Directions

When keeping in mind the difficulty of email action item agreement even between humans, we are pleased with the achieved *f*-score, especially considering that the current automatic annotation is only performed based on the classification model and rules. To construct our classifier we adopted a knowledge-based (KB) Information Extraction approach. The counterpart to this method is the Machine Learning (ML) approach. Our motivation was driven by the absence of sufficient gold standard training data but the availability of a language engineer.

In general KB systems are perceived as achieving high precision while ML approaches are more oriented towards higher recall. However, mixing both approaches is considered best practice. Thus we want to improve our classification by using the existing KB classifier to bootstrap an ML system, such that a gold standard training dataset can be generated from the rule-based annotations and subsequently corrected manually to ensure high quality. An ML-trained classifier will then be applied to annotations derived from the existing linguistic processing resources (tokenizer, POS tagger, NE lookup) to classify speech acts. The classifier will then only execute the existing rule-based approach as a fall back when it fails. We intend to employ the ML GATE plugin, which can load different ML engines such as the LibSVM package<sup>4</sup>, the PAUM algorithm (Zaragoza et al., 2002) or interface with Weka – an open source ML package (Witten & Frank, 1999).

We are confident that the addition of ML techniques will strengthen the relevance of the action items mined by Semanta. We also intend to extend the applicability of our technology to outline action items in Instant Messaging (IM) or electronic chats. There have already been a number of contributions in this area, mostly based on ML and statistical approaches. In particular, Fišer (2007) discussed ML techniques for the purpose of dialogue act recognition, whereas Forsyth and Martell (2007) built a chat corpus tagged with lexical, syntactic and discourse information; for classification via the use of statistical-based NLP. However, a problem which already affects email annotations, is more pronounced in IM. Chats are significantly less formal and the phenomenon of ‘text speak’ – the use of non-standard English words and language, limits the performance of our classification rules. Although the envisaged machine-learning support will partly solve this problem, we will investigate how the flexibility of our pipeline can be extended to handle this problem.

## 6. Acknowledgments

The work presented in this paper was supported (in part) by the Lion project supported by Science Foundation Ireland under Grant No. SFI/02/CE1/113.

## 7. References

- Carvalho, V., Cohen, W. (2006). Improving Email Speech Act Analysis via N-gram Selection. *In Proceedings of the Human Language Technology Conference / North American chapter of the Association for Computational Linguistics, ACTS Workshop*, New York City, NY.
- Comrie, B. (1985). *Tense*. Cambridge University Press.
- Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and

<sup>4</sup> A library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

- Applications. *In Proceedings of ACL'02*. Philadelphia, USA.
- Cunningham, H., Maynard, D., Tablan, K. (2000). JAPE: a Java Annotation Patterns Engine (2<sup>nd</sup> ed.). *Technical Report CS*, pp. 00--10. University of Sheffield, UK.
- Fišel, M. (2007). Machine learning techniques in dialogue act recognition. *Estonian Papers Applied Linguistics*.
- Forsythand, E.N., Martell, C.H. (2007). Lexical and Discourse Analysis of Online Chat Dialog. *In Proceedings of ICSC'07*. pp.19—26.
- Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., Kandola, J. (2002). The Perceptron Algorithm with Uneven Margins. *In Proceedings of the 9th International Conference on Machine Learning*. pp. 379--386.
- Moeschler, J. (1992). The pragmatic aspects of linguistic negation: Speech act, argumentation and pragmatic inference. *Argumentation*, 6(1), pp. 51—76.
- Ogihara, T. (2007). Tense and Aspect in Truth-conditional Semantics. *Lingua*, 117(2), pp. 392--418.
- Scerri, S., Davis, B., Handschuh, S., Hauswirth, M. (2009). Semanta – Semantic Email made easy. *In Proceedings of ESWC'09*. Crete, Greece.
- Scerri, S., Handschuh, S., Decker, S. (2008). Semantic Email as a Communication Medium for the Social Semantic Desktop. *In Proceedings of ESWC'08*. Tenerife, Spain.
- Scerri, S., Mencke, M., Davis, B., Handschuh, S. (2008). Evaluating the Ontology underlying sMail - the Conceptual Framework for Semantic Email Communication. *In Proceedings of LREC'08*. Marrakech, Morocco.
- Searle, J. (1969) Speech Acts. *Cambridge University Press*.
- Whittaker, S., Sidner, C. (1996). Email overload: exploring personal information management of email. *In Proceedings of CHI'96*. Vancouver, BC Canada, pp. 276--283.
- Witten, I.H., Frank, E. (1999). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. *Morgan Kaufmann*.