Auto-marking 2: An update on the UCLES-Oxford University research into using computational linguistics to score short, free text responses.

Jana Z. Sukkarieh[1], Stephen G. Pulman[1] and Nicholas Raikes[2]

[1] *Computational Linguistics Group, Centre for Linguistics and Philology, Walton Street, Oxford OX1 2HG, United Kingdom. Email: first.lastname@clg.ox.ac.uk*

[2] *Interactive Technologies in Assessment and Learning (ITAL) Unit, University of Cambridge Local Examinations Syndicate (UCLES[1]), 1 Hills Road, Cambridge CB1 2EU, United Kingdom. Email: N.Raikes@ucles-red.cam.ac.uk*

At last year's conference we introduced our on-going three year investigation into the automatic marking (scoring) of short, written answers of one or two sentences (Sukkarieh et al, 2003, henceforth SPR*). We described and gave initial results for preliminary implementations of two approaches to the automatic marking of some GCSE Biology questions. The first approach involved the use of information extraction techniques to extract significant features from answers and match them with patterns in a hand crafted marking way. The second approach used text classification techniques to match new, un-marked answers with the nearest equivalent answers in a sample of human-marked "training" answers. In the present paper, we report new results following refinements to the first approach, discuss problems encountered and describe progress towards our goal of automating the process of configuring the system to mark new questions. We finish by briefly discussing our priorities for future research and our plans for a real, low stakes application of auto-marking to be trialled during the coming year.

*Sukkarieh, J. Z., Pulman, S. G. & Raikes, N. *Auto-marking: using computational linguistics to score short, free text responses*. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK. October 2003.

[1] The UCLES Group provides assessment services worldwide through three main business units.

- Cambridge-ESOL (English for speakers of other languages) provides examinations in English as a foreign language and qualifications for language teachers throughout the world.
- CIE (University of Cambridge International Examinations) provides international school examinations and international vocational awards.
- OCR (Oxford, Cambridge and RSA Examinations) provides general and vocational qualifications to schools, colleges, employers, and training providers in the UK.

For more information please visit http://www.ucles.org.uk

Introduction

There is an old story about the student who says “This is what risk is all about” when handing in a blank exam paper to the examiners, having been asked to write an essay about taking risks. One examiner gives the student a full grade and the other gives a zero[1].

This is perhaps the limiting case of a test question where determining the intentions behind the answer cannot be achieved by natural language processing techniques. Fortunately, in automatic marking of short answers, we do not have to deal with this kind of case, but the still very difficult ones where the criteria of ‘right-or-wrong’ are relatively well-defined. For example, in Biology examinations like those we have been dealing with, aimed at 15-16 year olds, questions like the following illustrate our task:

> *Background:* *Cindy and Kathleen want to find out how quickly fly maggots use oxygen in comparison with woodlice. The students were given a chart and a table representing the results. An (x,y) coordinate graph is recorded, where x is time in mn (every 2 mn until 10mn) and y is the distance the coloured water moves in cm (the movement of the coloured water is an indicator on how much oxygen has been used by either maggots or woodlice) – the experiment is described on the exam paper but the last fact is not stated.)*

The Question

a. ***How*** *do the results with the maggots differ from the results with the woodlice?*

Possible Answers:

1. *the coloured water moves more*
2. *the coloured water moves faster*
3. *the slope of the graph in the table is steeper*

b. *Explain* ***why*** *the two sets of results are different*.

Possible answers (any 2)

1. *maggots use more oxygen or maggots use oxygen faster*
2. *maggots are more active or different rates of movement*
3. *different metabolic rates, different activity , using more energy or faster respiration*
4. *more maggots or more mass*

At last year’s conference we introduced the UCLES-Oxford University auto-marking project and described out first year’s work. The project is an initially three year investigation, funded by UCLES, into the application of computational linguistics to the marking (scoring) of short, free text answers to questions like the ones above.

In SPR we described how we set about investigating the application of Natural Language Processing (NLP) techniques to the marking of GCSE Biology answers of up to around five lines. We focussed on two techniques: an information extraction method, and a simple text classification method. We reported results for the two methods when marking some Biology questions. The work shares the same aims, and uses many similar techniques (although independently developed) as the systems described in Mitchell et al (2002, 2003), Rose at. al (2003) and Leacock et. al (2003). As far as it is possible to make sensible comparisons on systems that are not tested on the same data, all of this work achieves comparable levels of accuracy.

In the present paper we report on recent work trying to improve and refine the information extraction based system, with a particular focus on the issue of rapid porting to new questions and new domains. We briefly describe some techniques inspired by the machine learning

[1] This example is courtesy of Ms. Noha Kanj, an examiner on the Lebanese Baccalaureate.

literature. Work on the application of more sophisticated text classification methods, and machine learning techniques for these has also been continuing, but we will report that in a later paper.

In the following section, we remind the reader, briefly, on what information extraction is and that there are two ways to go about it, namely, a knowledge-engineering approach and a machine learning approach. In section 3, we first recapitulate the knowledge-engineering approach described in SPR. We report on some improvements, and a variant of the same general method, with some new results. In section 4, we describe one particular machine learning inspired approach and report some results for that.

From the outset we planned to produce a real, low stakes application of auto-marking for trialling in year three. We believe we will achieve this target, and in the final section of the paper we briefly describe our initial ideas.

Information Extraction in a Nutshell

Information extraction (IE) techniques pull out pertinent information from a partially syntactically analysed text by applying a set of domain specific patterns typically built from training data. In general, what defines pertinent information in such a technique varies from one subtask to another. Successful systems have been built for IE from semi-structured texts (like invoices or advertisements) and some more difficult texts (like news articles). In all these cases, the IE process is broken down into a series of subtasks. First, the named entity (NE) subtask consists of classifying (typically) NPs into categories like name of a person, a company, a location, or a date. Secondly, relations between named entities, or properties of entities are extracted. Thirdly, pre-defined templates are filled by these entities and relations.
Anaphora and ellipsis resolution may take place between these stages. In general, the information for filling a template may be found within a single sentence, across sequences of sentences, or sometimes in different forms several times within the same short text. Our auto-marking task is treated as information extraction from unstructured, pretty free text, and where the templates may be matched across sentences, and sometimes twice within the same passage.

The patterns or templates (we use the terms interchangeably here, although in some applications it makes sense to distinguish them) i.e., the rules that select from each text the information relevant to the task, are built from training data in either of the following ways. However, in either case we need to devise a language or a grammar to represent these rules.

Manually-Engineered Patterns

A person writes special knowledge to extract information using grammars and rules. The 3 crucial steps in which to write extraction rules by hand can be found, among other references on information extraction, in Appelt and Israel (1999). These, in order, are:

1. Determine all the ways in which the target information is expressed in a given corpus.
2. Think of all the plausible variants of these ways.
3. Write appropriate patterns for those ways.

This requires skill, a lot of labour, and familiarity with both domain and tools. To save time and labour various researchers have investigated machine learning approaches to learn IE patterns.

Automatic Pattern Learning

This approach requires a lot of examples with data to be extracted, and then the use of a suitable learning algorithm to generate candidate IE patterns. One family of methods for learning patterns requires a corpus to be annotated, at least to the extent of indicating which sentences in a text contain the relevant information for particular templates (Riloff 93, Soderland et. al 95, Huffman 95, Kim et. al 95, Califf 98). Once annotated, groups of similar sentences can be grouped together, and patterns abstracted from them. This can be done by taking a partial syntactic analysis, and then combining phrases that partially overlap in content, and deriving a more general pattern from them. `More general' here can mean either that e.g. we abstract away from syntactic and lexical variation so that `black dog', `dog that is black', `brown dog' etc all cluster as `X dog'; or that terms occurring in similar contexts (as arguments of the same class of verbs, for example) like `cat', `dog', etc. are all clustered together as instances of `animal'. This latter type of abstraction presupposes that some kind of semantic hierarchy or thesaurus is available. Alternative candidate abstractions can be tested for recall and precision on the training data so that only the most accurate are retained. This only requires people familiar with the domain to annotate text. However, it is still a laborious task.

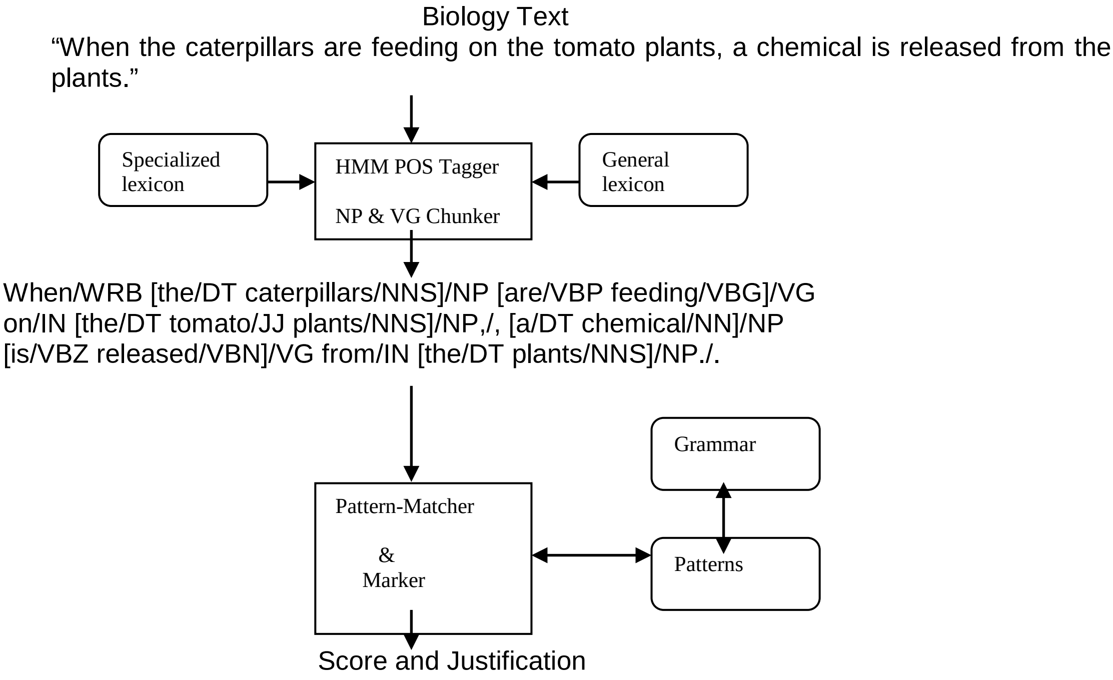
Another family of methods, more often employed for the named entity recognition stage, tries to exploit redundancy in un-annotated data (Riloff et. al 95, Rilof et. al 99, Yangarber 2001, Brin 98, Agichtein et. al 2000, Collins and Singer 99). One influential method (Collins and Singer, 1999) notes that the category of an NP is often signalled twice: e.g. in a phrase `Mr Vinken, president of Smith plc,…' the fact that `Mr. Vinken' is of category `person' is signalled both by the use of `Mr' (a substring of the NP) and the head of the appositive phrase, `president' (a `context' for the NP). Using a small set of initial hand crafted `seed' substring rules the initially unlabelled data is labelled. From this labelled data a set of `context rules' is derived. The data is relabelled using these rules, and the combined set of rules is scored in terms of how often they agree with each other. The highest scoring rules are added to the initial seed rules and the process is repeated. (This is known as `bootstrapping'). Collins and Singer report that the final system of rules was 90% accurate, although it should be noted that only four semantic categories were involved.

In the following section, we describe some preliminary results falling under the first approach, namely, the manually-engineered approach.

A Manually-Engineered Approach to Auto-Marking

Our Previous Work

Figure 1 illustrates the system we described in our last paper. We used a Hidden Markov Model part-of-speech (HMM POS) tagger trained on the Penn Treebank corpus, and a Noun Phrase (NP) and Verb Group (VG) finite state machine (FSM) chunker to provide the input to the information extraction pattern matching phase. The NP network was induced from the Penn Treebank, and then tuned by hand. The Verb Group FSM (i.e. the Hallidayean constituent consisting of the verbal cluster without its complements) was written by hand. Shallow analysis makes mistakes, but multiple sources help fill gaps and in IE it is adequate most of the time (in our previous paper we listed cases with detailed examples on where this fails).



Biology Text

"When the caterpillars are feeding on the tomato plants, a chemical is released from the plants."

When/WRB [the/DT caterpillars/NNS]/NP [are/VBP feeding/VBG]/VG on/IN [the/DT tomato/JJ plants/NNS]/NP,/, [a/DT chemical/NN]/NP [is/VBZ released/VBN]/VG from/IN [the/DT plants/NNS]/NP./.

Fig 1. Prototype 1 described in our last paper in IAEA 2003.

In our first attempt, given a question and answer, we try to identify a chunk of text in the answer that qualifies for a mark. We do this at a fairly concrete level on the basis of particular collections of keywords. After checking out the variant ways answers could be written, we devised a simple language in which to write the patterns:

```
Pattern        ->    Word | Word/Cat | Symbol
                     | Variable | Disjunction
                     | Sequence | k(N, Sequence)
                     (N is upper limit of length of Sequence)
                     | k(N, Sequence, Pattern)
                     (Sequence NOT containing Pattern)
Disjunction    ->    {Pattern, ..., Pattern}
Sequence       ->    [Pattern, ..., Pattern]
Word           ->    sequence of characters
Cat            ->    NN | VB | VG ...
Symbol         ->    & | % | $ ...
Variable       ->    X | Y | Z ...
```

It is easy then to build up named macros expressing more complex concepts, such as ‘negated verb group’, or ‘NP headed by word *protein*’, etc. Our rules centered around equivalence classes where the equivalence relation, R, is ‘…convey the same message/info as …” and these equivalence classes, in turn, centered around the marking scheme the examiners provided us with. We developed a pattern-matcher accordingly and the results on the unseen testing data did not deteriorate from the encouraging results on the training data (agreement with human markers for 88% of the answers). It should be clear that one can easily improve such results further by putting more work into the patterns.

The next step describes some refinements for this system since our last paper.

Our New Improved ‘Formula’

The system, in spirit, follows the same guidelines as in figure 1 above. However, it varies from the last system in many ways. In no particular order, these are:

1. The general-purpose lexicon now contains words with corresponding tags from the British National Corpus in addition to what we had from the Wall Street Journal.
2. The domain-specific lexicon obviously is an on-going process as we tackle more Biology questions. We are at the moment investigating adding a complete index from a GCSE Biology textbook with probabilities for a particular tag calculated depending on an on-line GCSE biology revision textbook.
3. Improvement of the finite state machine (FSM) chunker for finding verb groups (VG) and noun phrases (NP).
4. More specific NPs and VGs are now supported. They describe more detailed features like polarity, modifiers, semantics, etc. New types of features can be added as necessary.
5. The syntax of the grammar for writing patterns is different.
6. In the current grammar, the more complicated categories (like NP, VG, etc.) and their features are incorporated directly into the rules.
7. The combined pattern-matcher and marker shown in Figure 1 have been split into separate components of the system. Also, the patterns are now compiled before any pattern-matching or any marking is done.
8. We investigated using less than one third of the training data used previously to achieve similar marking accuracy.
9. We dealt with a novel type of marking scheme that we had not previously encountered.

We will now describe the new grammar, give a detailed example of how the matching is done and report some results from marking a few Biology questions.

A pattern takes the form:

Id :: LHS ==> RHS, where

Id can be a complex term to categorise patterns into groups and subgroups.

LHS ==> Cat, where

Cat is a (linguistic) category like NP, VG, Det, etc, or one that is user-defined.

RHS ==> [Element*] i.e. a list of elements, where

Element ==> Variable | Word/Cat | c(Cat)
|?(Element) optional element
| k(Sequence) | k(N, Sequence)
(Sequence of elements)
| (Element; Element) disjunction
W(Word)

Also each element could be followed by some condition. An example pattern is:

Id(1,N): complete ==> [c(vg:[head=V]), w(funny),?(w(weird)),w(M)],

Where V is a member of the set {encounter,come-across} and

M is a member of the set {example, biology}

The pattern therefore is a verb group headed by the words “encounter” or “come across”, followed by the word “funny” and optionally the word “weird”, followed by a word (M) which is either “case” or “biology”.

The pattern is abstracted over (i.e. decomposed into) all the individual patterns that fit:

Id(1,1): complete ==> [c(vg:[head=encounter]),w(funny),w(example)].

Id(1,2) : complete ==> [c(vg:[head=encounter]),w(funny),w(biology)].

Id(1,3): complete ==> [c(vg:[head=come-across]),w(funny),w(example)].

Id(1,4): complete ==> [c(vg:[head=encounter]),w(funny),w(weird), w(biology)], …, etc.

The first step in the pattern matching algorithm is that all patterns are compiled. Afterwards, when an answer arrives for pattern matching it is first tagged and all phrases (i.e. verb groups and noun phrases) are found. These are then compared with each element of each of the compiled patterns in turn until either a complete match is found or all patterns were tried and no match existed.

For example, consider the patterns above and the sample answer “we encounter funny people”.

First the input is tagged and the output passed to the pattern matching system.

The verb group “encounter” matches, so now the next element in the “encounter” patterns are looked for:

Found	Found where ? (Id)	Still looking for	To be what? (LHS)
Vg(encounter)	Question..pattern …	Funny,?(weird),example	Complete
Vg(encounter)	=	Funny,?(weird),biology	Complete
Vg(encounter)	=	Funny, example	Complete
Vg(encounter)	=	Funny, biology	Complete
Etc.			

The word “funny” matches, so the table becomes:

Found	Found where ?	Still looking for	To be what?
Vg(encounter), funny	Question..pattern …	?(weird),example	Complete
Vg(encounter), funny	=	?(weird),biology	complete
Vg(encounter), funny	=	example	complete
Vg(encounter),funny	=	biology	Complete
Etc.			

The process continues until a complete match is found or no match exists. In our example, no further elements will be matched since none of the words “funny”, “example” of “biology” exist in the input. If a complete match is found then the columns ‘still looking for’ to be ‘complete’ will eventually be empty for the row containing the pattern that was fully matched. This way the input is matched against all patterns before any marking is done. The marking process is very similar to the marking algorithm used in the first prototype (described in SPR).

We now report results from using the refined system to mark answers from the two Biology questions given near the beginning of the present paper. These are not the only questions we used for testing but for the purpose of this paper they will suffice to illustrate the issues (the results are in no way conclusive).

Results and Brief Discussion

The first question was:

> ***How** do the results with the maggots differ from the results with the woodlice?*

The second question was:

> *Explain **why** the two sets of results are different*.

Answers from handwritten scripts were keyed into a computer for the purposes of this research.

For the first question we used only 36 answers for writing the patterns. The resulting patterns scored 81% agreement with human examiners when we tested them on new answers. For the second question we used only 21 answers when pattern writing and achieved the same level of agreement (81%). The failures in both questions were mainly due to unanticipated alternatives for words or word-sequences that were included in the patterns. This, obviously, can be improved by looking at more answers when writing patterns (in the previous system we were looking on average at around 200 answers when pattern writing and now we looked on average at only 29). If we include these alternative wordings in the patterns, we achieve 90% agreement with examiners. As previously mentioned, with IE techniques one can easily improve the results by putting more effort into writing the patterns. Further, some mismatches were due to words marked as illegible by the transcribers but which could be deciphered by subject specialists (this problem would not apply in a real scenario where candidates typed their answers directly into a computer), or spelling or grammar errors in candidates’ answers (the handling of which is still under investigation).

It is worth noting:

1) The two questions used were problematic. Though one is asking 'how' and the other 'why' most students ignore the difference or write one for the other. This created some difficulty in deciding what to accept and what not to accept – human examiners are often lenient in these circumstances.

2) The amount of sample answers needed varies depending on a question's complexity which is often reflected in the number of marks available for the question.

3) We do not only look for positive answers that earn marks (i.e. right and relevant Biology), we also try looking for negative ones that prevent a mark from being awarded. For example, marking schemes sometimes specify answers that examiners should reject, and some answers contain wrong material that cancels out right material in the same answer (for example when a candidate has hedged his or her bets by giving contradictory alternatives in the same answer). The existence of wrong biology in an answer does not necessarily cancel out right biology and lead to 0 marks, however, so not all wrong biology is negative.

4) There will always be some answers that are impossible to mark automatically. The same is probably true for human examiners. The following two cases illustrate this point:

a. Handling spelling errors: The edit-distance algorithm that we have implemented is insufficient in recognizing and correcting some mis-spelled domain-specific words. Consider for example, the word 'miosis' that we found in one student's answer. The distance from 'miosis' to 'mitosis' or 'meiosis' is the same. It is hard to decide what to do in this case, even if the system provides a list of options the examiner/expert cannot decide what is intended!

b. Many anaphora resolution methods require sophisticated domain knowledge. A simple strategy based on grammatical parallelism would go wrong in the following example[2] (also it is impossible to know what the student intended):

> **Question:** why is the moon called the Earth's satellite?
> **A common answer by students** : 'it orbits it'.

The enhancements described above are quite promising but there is more overhead on the chunker, speller, etc for longer answers, especially those that have been written in a very ungrammatical way. We are in the process of checking the performance of the system without a tagger. We will report the results on a different occasion.

The process of designing a grammar for the patterns, writing patterns, testing the system, then improving/modifying the patterns (the well-known cycle in IE) is still laborious (though not as much as when we started since we have a bit more experience in the domain now). We therefore decided to investigate whether some machine learning (ML) methods could helps us semi-automate the process of writing patterns, and we now describe this work.

Automatic Pattern Learning for Auto-Marking

We do not have an annotated corpus in the usual sense, and we do not want to manually categorise answers into positive or negative instances since this is a laborious task. However, we do have a sample of human marked answers that have effectively been classified into different groups by the mark awarded. For example, for a 2 mark question these groups would be 0 marks, 1 mark, and 2 marks. Ideally every answer in the 0 marks group would have no positive instances, every answer in the 1 mark group would have some positive instances and possibly some negative ones, and every answer in the 2 mark group would contain no negative instances. In practice the situation is not always so clear cut and

[2] This example is courtesy of a Physics GCSE examiner we met at a coordination meeting among examiners.

this is the reason why the pattern learning process is so hard to automate. We also usually have some answer keys taken from the marking schemes written for human examiners. We can take aspects of the various ML techniques described in the literature and put them together. We want, as Catala et. al, “ a method that facilitates the process of building an IE system by limiting the expert’s effort to defining the task and supervising the … results [at each stage]” (Catala et. al 2003).

In the first subsection below we describe a baseline algorithm and in the second a semi-automatic algorithm. Afterwards, we report some results and discuss them briefly. The method we have at the moment is basically automating what we have been doing manually up to now.

Baseline

1. From the answer keys only, construct a set of sentences corresponding to correct answers.

2. Automatically produce common syntactic and lexical variants of these. For example,

The wasps parasitise the caterpillars => The caterpillars are parasitised by the wasps.
=> The insects parasitise the larvae etc.

This can be done using general linguistic pattern matching (effectively, the old style transformations of transformational grammar) + a thesaurus.

4. Apply the resulting patterns to the training corpus.

5. Tweak if necessary and go back to step 4, or

6. Apply the resulting patterns to the testing corpus.

A Supervised Learning or Semi-Automatic Algorithm

The aim, as previously mentioned, was to automate what we have been doing manually. Consider all answers whose mark is> 0 . We will leave answers whose mark is 0 for testing ‘abstract’ or general rules later.

1. Choose a set of keywords, i.e. all essential/salient keywords that occur in the marking scheme, and a set of synonyms or similar words for each keyword. Some words weigh more. As a rule of thumb, words (and their synonyms) that occur in the question (or are implied by the question – of course this last issue is not always easy to detect) are not considered essential and are initially weighted less than other words unless otherwise stated. The keywords are sorted into different equivalence classes (and if necessary into different sub-sets if keywords in different sets overlap).

   The choice of keywords and their weights is improved by practice and according to how many successful matches are found, i.e. by what we call the **learn-test-modify** iterative process and not the write-test-modify process of the manually-written approach to IE. Note, however, that modify in the learn-test-modify sequence does NOT mean modify the patterns but only modify the initial conditions of the algorithm.

2. Find the most specific positive 1-mark patterns:

   Having the set of keywords above,

   a. If a keyword can stand on its own like ‘vasoconstriction’ then there is nothing to learn – the pattern is exactly the individual word.

   b. Otherwise, choose, automatically, windows in the training answers around the keywords. The width of the window is chosen heuristically at the beginning yet again adjusted up or down through experience and the process of learn-test-modify.

   c. Classify each window as belonging to the same equivalence classes or sub-sets as the keyword it surrounds. If a keyword belongs to more than one set then this information might be necessary for decisions in step 3 below.

Ideally, there should be some restrictions on windows. For example:

   a. if a keyword is nominal and it appears as a subject in the marking scheme then pick a window of width=N to its right.

   b. if a keyword is nominal and it is an object then pick a window to its left.

   c. if it is a verb then let it be at the centre of the window, etc.

It is important to note here that the marking scheme provided does not consist of full sentences. We are not sure yet whether full sentences would be helpful or a hindrance.

3. Expert filtering:

   a. get rid of noise (i.e. completely irrelevant windows);

   b. remove unnecessary information from windows

4. The learning step (Generalisation or abstracting over windows):

   The patterns produced so far are the most-specific ones, i.e. windows of keywords only. We need some generalisation rules that will help us make a transition from a specific to a more general pattern. Starting from what we call a triggering window, the aim is to learn a general pattern that covers or abstracts over several windows. These windows will be marked as 'seen windows'. Once no more generalisation to the pattern at hand can be made to cover any new windows, a new triggering window is considered. The first unseen window will be used as a new triggering window and the process is repeated until all windows are covered (the reader can ask the authors for more details. These are left for a paper of a more technical nature).

5. Translate the patterns learned in step 4 into the syntax required for the marking process (if different syntax is used).

6. Expert filtering again for possible patterns.

7. Testing on training data. Make additional heuristics on width. Also, add or get rid of some initial keywords.

8. Testing on testing data.

The reader might think that since an expert is providing a list of similar words or expressions (including synonyms) for each keyword then it is unlikely that the algorithm will result in a lot of new variants. In practise, this is not the case. The reason is that an expert (even if it is a domain expert) gives general, what-occurs-to-mind alternatives but is unlikely to cover the myriad ways students express a certain concept in their answers. This is clearly the case from our observation of the coordination meetings of examiners. Using an ontology like WordNet may improve the initial keyword list or the synonym list for each keyword. However, we believe that large ontologies tend to, first, include rarer senses of words and phrases, which may throw the system off. Second, they tend to be too general. We believe that we need a more domain-specific context to look in and the best place to look for alternatives, synonyms or similarities is in the students' answers (i.e. the training data).

In the following section, we report the results of this algorithm when tried on answers to some Biology questions.

Results and Brief Discussion

The algorithm has been tested on only two Biology questions so far, namely:

1. The blood vessels help to maintain normal body temperature. Explain how the blood vessels reduce heat loss if the body temperature falls below normal.

   For which the marking scheme was:

   any three:
   vasoconstriction;
   explanation;
   less blood flows to / through the skin/close to the surface;
   less heat loss to air/surrounding/from the blood/less radiation/conduction/convection;

   REJECT blood vessels moving
   REJECT smaller
   IGNORE reference to hairs
   IGNORE shivering

2. If the body gets cold, it attempts to increase its temperature. Write down one way it does this.

   Mark scheme:

   any one:
   shivering / increased liver activity / less sweating/increased metabolic rate/hairs standing up/vasoconstriction or explanation;

   IGNORE put on clothes

These questions were chosen, firstly, because we had already manually written patterns for them and we wanted to compare the hand-crafted patterns with the learned patterns (the comparison was done manually). The second, more important, reason we chose these two questions was that one seems to be a subpart of the other. In other words, the answer for question (1) could be used as an answer for question (2). This was probably unintended by the question writers when they wrote it, and if we were using the question in a real system we would have revised it, but for the purposes of checking our algorithm we wanted to use challenging examples. For the above two questions we expected that patterns learned from students' answers to the first question were going to occur again in answers for question (2). However, not all patterns in question (1) are acceptable for question (2) since, for example, the mark scheme does not allow "less heat loss" for question (2). This means that the 'right' parts of students' answers for question (1), that are not allowed for question (2), were considered, by the learning process, to be positive instances for question (1) yet negative instances for question (2). This seemed challenging enough for our learning process.

We found about 89% agreement among words between the learned patterns and the hand-written patterns, a promising result for this challenging sample of data. However, it is worth noting a few issues, though this is not an exhaustive list of the issues encountered so far (we excluded a lot of details in the algorithm and hence in the corresponding discussion). First, the fact that we had already written patterns manually for these questions may have helped us, as experts and knowledge engineers, when defining the initial conditions of the algorithms. The obvious next step is to test it on completely new questions and data for a better indication of the results and agreement with examiners. Second, in the learning step (step 4 in the algorithm above), we defined a default priority order among features that unfortunately sometimes results in an attempt to learn the wrong linguistic feature in an answer. For example, to learn from ‘less heat loss’, ‘minimum heat loss’, ‘reduced heat loss’, ‘decreased heat loss’ the priority should be for the modifiers of the NP and not the head of the NP as our default priority order would have it. Finally, some patterns were not learned simply because we haven’t yet implemented support for them. For example, ‘less blood flows through the skin’, ‘less blood flows close to the skin’, ‘less blood flows to the skin’, etc, were not abstracted over since we haven’t implemented support for the case where the preposition(s) in a prepositional phrase is possibly what we are looking for. Different phases of testing of the algorithm are in progress, and the results of these tests will give us a clearer idea of the refinements we need to make to this first implementation of the **learn-test-modify** process.

Summary and future work

In this paper we have described some refinements made to a prototype system described in our paper at last year’s IAEA conference (SPR). We have focused on information extraction techniques and have described algorithms for creating patterns manually and semi-automatically. The results for both algorithms are quite promising.

Further research

We will continue refining the methods described in this paper. We are also investigating other machine learning techniques, including the classification methods described in SPR. We are looking at question-answering (Q/A) research (that is, research into systems that automatically answer questions asked of them in free text by findings specific answers in a collection of free text documents). In particular, we are looking at the problem of judging possible answers returned. Though the kind and domain of questions dealt with in the Q/A literature up to now are different from ours, we do share the same concerns. The Q/A community haven’t found the solution to this problem yet, but we may benefit from some aspects in the way they handle question-typing, answer identification, ranking, confidence measure assignment, etc. Last but not least, though we have incorporated some basic semantic features into our existing system we would like to incorporate some deeper semantic features in order to predict, more reliably, acceptable variations of an answer.

Trialling auto-marking for low stakes assessment

From the outset we planned to produce a real, low stakes application of auto-marking for trialling in year three. Results from our research so far have convinced us that this target is achievable – we already know enough to produce a system that will probably mark short answers well enough for low stakes use. Further research may continue, but a major objective for the coming year is to wrap up what we know into a usable system for trialling.

Planning is still at an early stage, but the following are likely to be features of our prototype application:

- Short, factual answers suitable for auto-marking will be used in an assessment involving other item types (e.g. multiple choice, hotspot selection) suitable for conventional machine scoring. Some teacher marked items may also be used.

- The auto-marking engine will be integrated with an existing Computer Based Testing (CBT) system – there are plenty available and we do not want to re-invent the wheel, and our clients are unlikely to want to adopt completely new systems just so they can use our auto-marking system.

- The auto-marking engine will just mark short answer questions. It will not interact directly with students or mark other item types. The aim is that when the CBT system has a short answer for marking it will send it to the auto-marking engine, the engine will mark it and return the result for the CBT system to process as appropriate.

Possible contexts for the application include:

OCR GCSE Science Computer Based Testing

OCR is introducing low stakes tests for GCSE students that are purely computer based and computer marked. The tests will initially consist of graphical and multiple choice items. Could we integrate the auto-marking engine with this system and introduce some automatically marked items involving short, written answers?

CIE Geography Assessments

CIE has been investigating the use of simulations in, amongst other applications, Geography assessments (Puntis et al, 2003), (Puntis and Maughan, 2004). Assessment activities are built around simulations of a practical investigation. Students manipulate the simulation and answer questions on screen, some of which involve short, textual answers. A mixture of computer and human marking is used, with humans marking the items unsuited to conventional computer scoring techniques. Could we integrate the auto-marking engine with this system and try automatically marking some of the short, written answers?

Bibliography

Agichtein, E. and Gravano, L. (2000) Snowball: Extracting relations from large plaintext collections. Proceedings 5th ACM International Conference on Digital Libraries.

Appelt, D. and Israel, D. (1999) Introduction to Information Extraction Technology. IJCAI 99 Tutorial.

Brin, S. (1998) Extracting patterns and relations from the World Wide Web. WebDB Workshop at 6th international Conference on Extending Database Technology (EDBT'98), pp. 172-183.

Califf, M.E. (1998) Relational Learning Techniques for Natural Language Information. PhD thesis, Artificial Intelligence Laboratory, University of Texas at Austin.

Catala, N. and Castell N. (2003) A portable method for acquiring information extraction patterns without annotated corpora. Natural Language Engineering 9(2): 151-179. Cambridge University Press.

Collins, M. and Singer, Y. (1999) Unsupervised models for named entity classification. Proceedings Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Corpora, pp. 189-196.

Huffman, S. B. (1995) Learning information extraction patterns from examples. In: S.Wermter, E. Riloff and G. Sheler, editors, Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, pp. 246-260. Springer-Verlag.

Kim, J.-T. and Moldovan, D. I. (1995) Acquisition of linguistic patterns for knowledge-based information extraction. IEEE Trans. Knowl. And Data Eng. 7(5): 713-724.

Leacock, C. and Chodorow, M. (2003) C-rater: Automated Scoring of Short-Answer Questions. Computers and Humanities 37:4.

Mitchell, T. Russel, T. Broomhead, P. and Aldridge, N. (2002) Towards robust computerized marking of free-text responses. In 6th International Computer Aided Assessment Conference, Loughborough.

Mitchell, T. Russel, T. Broomhead, P. and Aldridge, N. (2003) Computerized marking of short-answer free-text responses. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

Puntis, A. Maughan, S and Beedle, P. (2003) Assessing New Educational Goals. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

Puntis, A. and Maughan, S. (2004) Using Simulations to Assess New Educational Goals. Paper presented at the 30th annual conference of the International Association for Educational Assessment (IAEA), Philadelphia, Pennsylvania USA.

Riloff, E. (1993) Automatically constructing a dictionary for information extraction tasks. Proceedings 11th National Conference on Artificial Intelligence, pp. 811-816.

Riloff, E. and Jones R. (1999) Learning dictionaries for information extraction by multi-level bootstrapping. Proceedings 16th National Conference on Artificial Intelligence (AAAI-99), pp. 474-479.

Riloff, E. and Shoen, J. (1995) Automatically acquiring conceptual patterns without an annotated corpus. Proceedings Third Workshop on Very Large Corpora, pp. 148-161.

Rose, C. P. Roque, A., Bhembe, D. and VanLehn, K. (2003) A hybrid text classification approach for analysis of student essays. In Building Educational Applications Using Natural Language Processing, pages 68-75.

Soderland, S., Fisher, D., Aseltine, J. and Lenhert, W. (1995) CRYSTAL: Inducing a conceptual dictionary. Proceedings 14th International Joint Conference on Artificial Intelligence, pp. 1314-1321.

Sukkarieh, J. Z., Pulman, S. G. & Raikes, N. *Auto-marking: using computational linguistics to score short, free text responses*. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

Yangarber, R. Grishman, R. Tapanainen, P. and Huttunen, S. (2000) Automatic acquisition of domain knowledge for information extraction. Proceedings 18th international Conference on Computational Linguistics (COLING 2000), pp. 940-946.