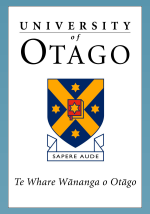
Tauira: A tool for acquiring unknown words in a dialogue context

Maarten van Schagen
Human Media Interaction Group
University of Twente

Alistair Knott
Dept of Computer Science
University of Otago



Outline of the talk

1. Background: the Kaitito dialogue system.
2. The problem of unknown words, and some current approaches to solving it.
3. Our solution: **dialogue-based word authoring**.
4. Some examples of the authoring system.

The Te Kaitito project

We wanted to build a NLP system which has ‘a little bit of everything’.

The Te Kaitito project

We wanted to build a NLP system which has ‘a little bit of everything’. We chose to build a **dialogue system**, as this requires a wide range of different components.

The Te Kaitito project

We wanted to build a NLP system which has ‘a little bit of everything’. We chose to build a **dialogue system**, as this requires a wide range of different components.

- sentence interpretation, sentence generation

The Te Kaitito project

We wanted to build a NLP system which has ‘a little bit of everything’. We chose to build a **dialogue system**, as this requires a wide range of different components.

- sentence interpretation, sentence generation
- anaphora, presuppositions, discourse structure

The Te Kaitito project

We wanted to build a NLP system which has ‘a little bit of everything’. We chose to build a **dialogue system**, as this requires a wide range of different components.

- sentence interpretation, sentence generation
- anaphora, presuppositions, discourse structure
- questions and answers

The Te Kaitito project

We wanted to build a NLP system which has ‘a little bit of everything’. We chose to build a **dialogue system**, as this requires a wide range of different components.

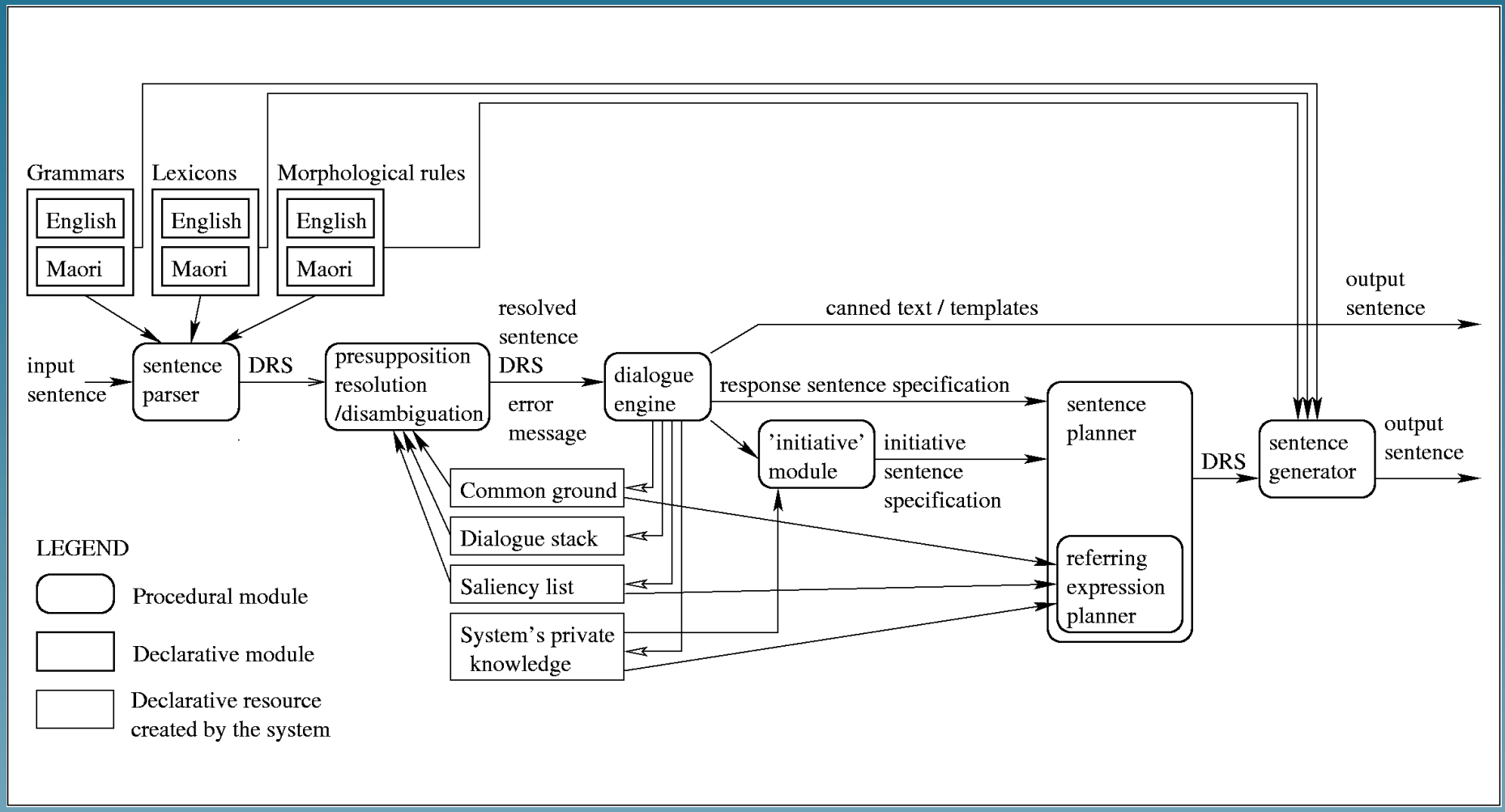
- sentence interpretation, sentence generation
- anaphora, presuppositions, discourse structure
- questions and answers
- information structure and intonation

The Te Kaitito project

We wanted to build a NLP system which has 'a little bit of everything'. We chose to build a **dialogue system**, as this requires a wide range of different components.

- sentence interpretation, sentence generation
- anaphora, presuppositions, discourse structure
- questions and answers
- information structure and intonation
- dialogue management issues

Architecture of the dialogue system



A simple one-on-one dialogue

A simple one-on-one dialogue

User: A black dog was in a garden.

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
User:  The dog barked.
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
User:  The dog barked.
TK:    Wait---which dog barked?
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
User:  The dog barked.
TK:    Wait---which dog barked?
User:  The white dog.
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
User:  The dog barked.
TK:    Wait---which dog barked?
User:  The white dog.
TK:    Okay. Okay.
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
User:  The dog barked.
TK:    Wait---which dog barked?
User:  The white dog.
TK:    Okay. Okay.
User:  Which dog was chased by the white dog?
```

A simple one-on-one dialogue

```
User:  A black dog was in a garden.
TK:    Okay.
User:  A white dog chased the dog.
TK:    Okay.
User:  The dog barked.
TK:    Wait---which dog barked?
User:  The white dog.
TK:    Okay. Okay.
User:  Which dog was chased by the white dog?
TK:    The black dog.
```

A simple one-on-one dialogue

User: A black dog was in a garden.
TK: Okay.
User: A white dog chased the dog.
TK: Okay.
User: The dog barked.
TK: Wait---which dog barked?
User: The white dog.
TK: Okay. Okay.
User: Which dog was chased by the white dog?
TK: The black dog.
User: I roto tēhea kurī i te māra? *What was in the garden?*

A simple one-on-one dialogue

User: A black dog was in a garden.
TK: Okay.
User: A white dog chased the dog.
TK: Okay.
User: The dog barked.
TK: Wait---which dog barked?
User: The white dog.
TK: Okay. Okay.
User: Which dog was chased by the white dog?
TK: The black dog.
User: I roto tēhea kurī i te māra? *What was in the garden?*
TK: Nā te kurī pango. *The black dog.*

The problem of unknown words

Unknown words are a big problem for computational grammars. E.g: when testing the ERG grammar on an unseen corpus, unknown words accounted for 40% of unparsed sentences (Baldwin *et al.*, 2004).

The problem of unknown words

Unknown words are a big problem for computational grammars. E.g: when testing the ERG grammar on an unseen corpus, unknown words accounted for 40% of unparsed sentences (Baldwin *et al.*, 2004).

Two approaches to unknown words:

- **Off-line**: increasing the size of the lexicon.
- **On-line**: interpreting an unknown word during parsing.

This talk is concerned with on-line methods.

Word-authoring dialogues

A dialogue application allows some interesting on-line methods for dealing with unknown words: the system can simply **ask the user** about the unknown word.

Word-authoring dialogues

A dialogue application allows some interesting on-line methods for dealing with unknown words: the system can simply **ask the user** about the unknown word.

We need a word-authoring dialogue which is

1. accurate;
2. manageable by non-linguists.

Knight (1996): unknown word syntax

Knight proposes embedding an unknown word in **example syntactic contexts** to determine its syntactic properties.

```
U:  John is hungry.
```

Knight (1996): unknown word syntax

Knight proposes embedding an unknown word in **example syntactic contexts** to determine its syntactic properties.

```
U:  John is hungry.
S:  I don't know the meaning of 'hungry'.
    Is 'very hungry' a reasonable phrase?
```

Knight (1996): unknown word syntax

Knight proposes embedding an unknown word in **example syntactic contexts** to determine its syntactic properties.

```
U:  John is hungry.
S:  I don't know the meaning of 'hungry'.
    Is 'very hungry' a reasonable phrase?
```

Problem: how do you choose the example phrases?

Knight (1996): unknown word semantics

Knight proposes two methods for determining an unknown word's semantics. Firstly via questions to situate a word in an existing type hierarchy:

```
U:  John is hungry.
```

Knight (1996): unknown word semantics

Knight proposes two methods for determining an unknown word's semantics. Firstly via questions to situate a word in an existing type hierarchy:

```
U:  John is hungry.
S:  Is 'hungry' a visually detectable property?
```

Knight (1996): unknown word semantics

Knight proposes two methods for determining an unknown word's semantics. Firstly via questions to situate a word in an existing type hierarchy:

```
U:  John is hungry.
S:  Is 'hungry' a visually detectable property?
```

Problem: it's hard for non-linguists to answer such questions.

Knight (1996): unknown word semantics

A second approach to asking about a word's meaning is via a paraphrase test:

```
U:  John is hungry.
```

Knight (1996): unknown word semantics

A second approach to asking about a word's meaning is via a paraphrase test:

```
U:  John is hungry.
S:  Can you paraphrase that?
```

Knight (1996): unknown word semantics

A second approach to asking about a word's meaning is via a paraphrase test:

```
U:  John is hungry.
S:  Can you paraphrase that?
U:  John wants to eat.
```

Knight (1996): unknown word semantics

A second approach to asking about a word's meaning is via a paraphrase test:

```
U:  John is hungry.
S:  Can you paraphrase that?
U:  John wants to eat.
```

Approach here: define words in terms of other words.

Deriving syntactic information from context

The sentence in which an unknown word appears provides lots of information about its syntactic properties.

Deriving syntactic information from context

The sentence in which an unknown word appears provides lots of information about its syntactic properties.

Some useful assumptions:

- The word is of a syntactic type known to the grammar.
- If the word were identified as being of this type, the sentence would parse.

Deriving syntactic information from context

The sentence in which an unknown word appears provides lots of information about its syntactic properties.

Some useful assumptions:

- The word is of a syntactic type known to the grammar.
- If the word were identified as being of this type, the sentence would parse.

From these we can derive a set of candidate word types.

Barg & Walther (1998)

Barg & Walther develop a system for processing monologues, in which the set of candidate types for each unknown word is open to constant revision.

- The first occurrence of the word creates a set of hypotheses.
- Subsequent occurrences of the word can reduce this set.

(Of course, this relies on seeing the right subsequent sentences.)

Barg & Walther (1998)

Formulating hypotheses is complicated by the presence of **syntactic features**.

- **Generalisable** features: can have different values in different contexts. E.g. `gender` for *child*.
- **Specialisable** features: always take the same values. E.g. `numbder` for *child*.

Fouvry (2003)

Fouvry adapts Barg & Walther's algorithm. He notes a useful simplification:

- There's a trend in unification-based grammars towards getting rid of features, by compiling them into the hierarchy of lexical types.
  E.g. `noun`, `masc-noun`, `fem-noun`.

Now the hypotheses about the syntactic properties of an unknown word can simply be a list of lexical types.

Some limitations of context-based methods

Some questions that no-one has considered yet:

Some limitations of context-based methods

Some questions that no-one has considered yet:

- How to handle sentences containing more than one unknown lexeme.

Some limitations of context-based methods

Some questions that no-one has considered yet:

- How to handle sentences containing more than one unknown lexeme.
- How to handle an unknown multi-word lexeme.

Some limitations of context-based methods

Some questions that no-one has considered yet:

- How to handle sentences containing more than one unknown lexeme.
- How to handle an unknown multi-word lexeme.
- How to handle inflected unknown words.

Some limitations of context-based methods

Some questions that no-one has considered yet:

- How to handle sentences containing more than one unknown lexeme.
- How to handle an unknown multi-word lexeme.
- How to handle inflected unknown words.
  E.g. Consider the sentence $I\ bobsled\ldots$

Summary

Dialogue-based and context-based approaches to unknown words are complementary:

- There are neat methods for determining a set of hypotheses about an unknown word from the sentences where it appears.

Summary

Dialogue-based and context-based approaches to unknown words are complementary:

- There are neat methods for determining a set of hypotheses about an unknown word from the sentences where it appears.
  But processing a set of sentences in 'batch' mode is a rather undirected way of reducing this set.

Summary

Dialogue-based and context-based approaches to unknown words are complementary:

- Asking the user questions about possible contexts for the unknown word should be able to narrow down the set of hypotheses fast.

Summary

Dialogue-based and context-based approaches to unknown words are complementary:

- Asking the user questions about possible contexts for the unknown word should be able to narrow down the set of hypotheses fast.
  But how do we decide which questions to ask?

Proposal (1): using test suites to generate questions

The sentences in the grammar's **test suite** can be used to generate good word-authoring questions.

- All wide-coverage grammars nowadays come with a purpose-built test suite.
- If the test suite is good, it should contain sentences which exercise all the different features of the grammar.
- It will also contain useful **minimal pairs** of sentences.

Proposal (2): multiple unknown words

In a dialogue context, we can get around the problems raised by multiple unknown words by asking the user some simple questions.

- For consecutive unknown words: *Are these two words part of the same lexical item, or separate?*
- For separate lexical items: *Can you please enter a sentence which uses just one of these items?*

Proposal (3): inflected unknown words

Try decomposing the unknown word using each applicable **morphological rule**.

- For each successful rule application, add the tuple `<lex-type,word-stem,morph-rule>` to the set of hypotheses being maintained.

Proposal (3): inflected unknown words

Try decomposing the unknown word using each applicable **morphological rule**.

- For each successful rule application, add the tuple <lex-type,word-stem,morph-rule> to the set of hypotheses being maintained.

E.g. from *The winner is zapfing*, we derive
{<v,*zapf*,-ing>,<v,*zapfe*,-ing>,<pn,*zapfing*,nil>}

Proposal (4): multilingual paraphrases for word semantics

Once the unknown word's syntactic type is established:

- We give it a 'null' semantics, then parse the original sentence and derive its semantics.
- Then we ask the user for a translation of this sentence, parse this, and derive its semantics.
- The semantics of the new word is the set of all predicates in the translation but not in the original sentence.

The Tauira system

The Tauira system

If the parser encounters an unknown word during a dialogue, Tauira initiates a clarification subdialogue.

The Tauira system

If the parser encounters an unknown word during a dialogue, Tauira initiates a clarification subdialogue.

- It begins with questions about multiple unknown words (as just described).

The Tauira system

If the parser encounters an unknown word during a dialogue, Tauira initiates a clarification subdialogue.

- It begins with questions about multiple unknown words (as just described).
- When it has a sentence with a single unknown word, it asks a series of questions about its syntax.

The Tauira system

If the parser encounters an unknown word during a dialogue, Tauira initiates a clarification subdialogue.

- It begins with questions about multiple unknown words (as just described).
- When it has a sentence with a single unknown word, it asks a series of questions about its syntax.
- Then it asks for a translation of the sentence, to establish

its semantics.

Creating test sentences from the test suite

We preprocess the test suite offline to produce a set of **test items**. A test item has the following fields:

- A sentence from which one 'target word' has been extracted.
- The original lexical type and morph-rule of this word.
- The set of all other lexical types which the grammar would allow to appear in the position of the target word.

Some example test items

From the test suite sentence *How happy was Abrams* we can derive the following test items:

- *How _ was Abrams*, `<adj,nil>`, {adjective,adverb}
- *How happy was _*, `<pn,nil>`, {pn,dayoftheweek,. . . }

Some example test items

From the test suite sentence *How happy was Abrams* we can derive the following test items:

- *How _ was Abrams*, `<adj,nil>`, {adjective,adverb}
- *How happy was _*, `<pn,nil>`, {pn,dayoftheweek,. . . }

We can now evaluate formally how good the test set is at providing contexts which distinguish between the different lexical types.

Evaluating the set of test items

For each lexical type t, we can compute:

1. The number of **attestations** of t in the test set:

- I.e. the number of test-items whose original type is t.
- If this is zero, there are no good sentences the system can use to verify if a word is of this type.

Evaluating the set of test items

For each lexical type t, we can compute:

2. The number of **positively indistinguishable** word types for t:

- I.e. the number of types found in *all* test items which include t.
  I.e. the number of hypotheses which would remain if the user affirmed that an unknown word could be used in all of these items.

Evaluating the set of test items

For each lexical type t, we can compute:

3. The number of **negatively indistinguishable** word types for t:

- Type t_2 is negatively distinguishable from t if there is a test item where t can be used and t_2 *cannot* be used.
- If the user says that the new word is okay in this test item, we know it's not of type t_2.

Questioning strategies

If the unknown word is morphologically unambiguous, we generate **multiple choice** questions, as follows:

Questioning strategies

If the unknown word is morphologically unambiguous, we generate **multiple choice** questions, as follows:

- Pick the commonest type from the set of hypotheses.

Questioning strategies

If the unknown word is morphologically unambiguous, we generate **multiple choice** questions, as follows:

- Pick the commonest type from the set of hypotheses.
- Select the test item for this type whose associated types has the smallest intersection with the hypothesis types.

Questioning strategies

If the unknown word is morphologically unambiguous, we generate **multiple choice** questions, as follows:

- Pick the commonest type from the set of hypotheses.
- Select the test item for this type whose associated types has the smallest intersection with the hypothesis types.
- Then iterate, choosing additional test items whose set of associated types do not intersect with those of the items chosen so far.

Questioning strategies

For unknown words with morphological ambiguity, **yes-no** questions are generated about the validity of single sentences.

Finally, if the test suite simply fails to provide sentences which distinguish between some of the candidate types, we fall back on a simple **similar words** question, presenting examples of words of the different types.

An example

An example

```
U:  I danced
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
U:  no
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
U:  no
S:  Which of these words is most like your word ?
    1. 'stand', 'orientate', 'bay'
    2. 'bust', 'grow', 'blow'
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
U:  no
S:  Which of these words is most like your word ?
    1. 'stand', 'orientate', 'bay'
    2. 'bust', 'grow', 'blow'
U:  1
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
U:  no
S:  Which of these words is most like your word ?
    1. 'stand', 'orientate', 'bay'
    2. 'bust', 'grow', 'blow'
U:  1
S:  Okay.
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
U:  no
S:  Which of these words is most like your word ?
    1. 'stand', 'orientate', 'bay'
    2. 'bust', 'grow', 'blow'
U:  1
S:  Okay.
U:  John dances.
```

An example

```
U:  I danced
S:  I do not understand the word DANCED.
    Is 'the dog arrived and danced' a correct sentence ?
U:  yes
S:  Is 'abrams dancs for browne' a correct sentence ?
U:  no
S:  Which of these words is most like your word ?
    1. 'stand', 'orientate', 'bay'
    2. 'bust', 'grow', 'blow'
U:  1
S:  Okay.
U:  John dances.
S:  Okay.
```

A multi-lexeme, multi-lingual example

U: the albatross chases the yelloweyed penguin
S: I do not understand the words ALBATROSS, YELLOWEYED, PENGUIN.
Is YELLOWEYED PENGUIN one word ?
U: yes
S: Could you provide me an example sentence with the unknown word 'ALBATROSS' using words I know ?
U: the albatross eats
S: Which of these phrases illustrates the correct use of ALBATROSS ?
1. the albatrosses of the projects are trustworthy
2. mr albatross browne is the manager
3. browne was hired on january albatross 1984
4. None
U: 4

A multi-lexeme, multi-lingual example

S: Is 'he showed her an albatross' a correct sentence ?
U: yes
S: Which of these words is most like your word ?
1. 'sec', 'counter narcotics', 'paycheck'
2. 'school', 'black', 'distance'
3. 'delivery', 'record', 'treatment'
4. 'specification', 'subscriber', 'deviation'
U: 1
S: What is the translation of 'the albatross eats' in Māori.
U: ka kai te toroa
S: Is 'kua pai te toroa' a correct sentence ?
U: yes

A multi-lexeme, multi-lingual example

S: Which of these phrases illustrates the correct use of YELLOWEYED PENGUIN ?
1. the yelloweyed penguins of the projects are trustworthy
2. mr yelloweyed penguin browne is the manager
3. browne was hired on january yelloweyed penguin 1984
4. there are five yelloweyed penguin in the room aren't there
5. None

U: 5

S: Is 'list yelloweyed penguins who have bookcases' a correct sentence ?

U: yes

A multi-lexeme, multi-lingual example

S: Which of these words is most like your word ?
1. 'sec', 'counter narcotics', 'paycheck'
2. 'school', 'black', 'distance'
3. 'delivery', 'record', 'treatment'
4. 'specification', 'subscriber', 'deviation'

U: 1

S: What is the translation of 'the albatross chases the yelloweyed penguin' in Māori.

U: ka whai te toroa i te hoiho

S: 'the albatross chases the yelloweyed penguin' translates to:
ka whāia te hoiho e te toroa
ka whai te toroa i te hoiho

For more on Te Kaitito, see

`http://tutoko.otago.ac.nz:8080/teKaitito/`