Word Prediction in SAW 6

Existing predictors from SAW 5

SAW 5 could use several different prediction systems: Penfriend, WordAid and Prophet. These each needed to be installed separately and SAW requested predictions from them as needed.
All of these are fairly old programs and have issues on more recent systems. Windows 7 causes some problems, and as far as I know none of these will correctly track keys pressed on 64-bit machines.
The code to support these has not been changed in SAW 6. Therefore if they are already working on a machine with saw SAW 5 they should also work in SAW 6. However no attempt has been made to update these and they will probably progressively become less useful as 64-bit and Windows 7 or later become more common.

Prediction in SAW 6

Because the publication with external systems had always been a bit flaky, the decision was made to add proper word prediction within SAW 6 itself. This is a completely new word prediction engine (called "Blade"). This is installed as part of SAW 6, and runs within SAW 6 not as a separate program.
It does however reside in a separate open source DLL, so other software can be written to use it.

Functionality

The new prediction engine has the following functionality:

- Records the frequency of each word
- Records the frequency of word pairs, triplets and larger groups (it doesn't remember every combination of words - this would require and almost unlimited amount of memory. It decides automatically what is likely to have an impact on the predictions it makes)
- Updates its information as the user types (in fact at the end of each sentence)
- Predicts differently at the beginning of a sentence. Both the first word is modified - predicting words which commonly appear at the beginning of a sentence, and also the next few words can take into account the position of the sentence start.
- Tracks words recently typed and moves them up the prediction list. (Because when typing an email or document, some words or names relevant to the subject of the document are likely to recur even though they might be quite rare in the language overall)
- Takes account of other punctuation and numbers in deciding the most likely next word (not just full stop which triggers predictions for the start of a sentence)
- In SAW: can track letters both typed using the machine's keyboard as well as within a SAW keyboard.
- Learns which words have capital letters - will predict words using capital letters if they are usually typed with capitals, when not at the beginning of a sentence.

In addition Blade offers optional **abbreviation expansion**:

- The list of abbreviations must be set up by the user – the system installs without any pre-defined abbreviations in any language. The abbreviation list consists of a list of abbreviations and the text into which they expand. Whenever one of the abbreviations is typed it is immediately expanded. If there should be a trigger character at the end of the abbreviation (eg abbreviations are expanded on pressing “.”, then this must be included in the defined abbreviations). The abbreviation must be preceded by either a space or new line to be processed. No preceding space should be included in the definition.

- The abbreviations can be defined in the user vocab editing screen. In SAW this can be launched via Menu > View > View word prediction words...

There is a separate document explaining the above listed functionality in more technical detail, and how Blade works, mainly intended for programmers.

Converting scripts to use Blade

Which word prediction engine is used by SAW is determined by the selection set startup script. This will contain a line to initialise word prediction, such as:

```
wordlist set "prophet|prediction"  2921
```

The number on the end is the ID of the item where the predictions will be displayed. The quoted part in the middle specifies the engine.
The new word prediction is initialised in exactly the same way, but using "blade|" - note that SAW requires a | character within the quotes. The engine name is before this character, and any options after it. Therefore to use the new word prediction system, the above line would be changed to:

```
wordlist set "blade|"  2921
```

Settings

Blade has some optional settings which can also be specified in this command. The settings go inside the quotes after the | character:

```
wordlist set "blade|AlphabeticalResults=true,Uppercase=true"  2921
```

The individual settings are separated by commas, and each is in the form "setting=value". They are not case-sensitive.
The settings can also be changed at any time when the script is running using this new script command:

```
blade settings "settings"
```

(when adding a script command this is listed at the end, in the miscellaneous section. There is no editing wizard, it is necessary to Edit Mode in SAW to edit this)

The possible settings are:

AlphabeticalResults = *true* or *false*
If this is set to true, the predictions are returned in alphabetical order. If false, they are listed most likely first (this is the default)

OmitPreviousSuggestions = *true* or *false*
If *true* then predictions will not include those which were suggested when one character less had been typed. So if after just pressing 'g' the predictions included "going", and this was not selected, it will not be predicted again if you then type 'o'. Presumably if it wasn't the right word the first time, then it still won't be the intended word.
A word is only omitted after the next character is typed. It will be displayed again if it still matches after 2 characters have been typed. Thus in the above example after typing 'i', blade will now predict "going" again.
The default for this setting is *true*

SimpleSingleLetter = *true* or *false*

If this is set to true, then after a single letter at the beginning of a word has been typed, Blade always gives the same predictions for each possible letter. I.e. it does not adjust the predictions based on the previous words. The predictions will simply be the most common words starting with the letter.
The purpose is that if users know what predictions will be generated, they can select them more rapidly, and the benefit of this may outweigh the fact that the intended word is less frequently predicted.

MinimumLength = *N* (a number)
This sets the minimum length for predicted words. Any word with fewer than N characters will not be included in the predictions. The idea is that for some users there is little point predicting some very short but common words (e.g. "a") because they can just as quickly be typed.
The default value is 0 which includes all words in the available data.

MinimumGain = *N* (a number)
This is similar to MinimumLength, but the limit is based upon the number of characters already typed. This specifies that only words containing N extra characters, beyond those which have already been typed, should be included. Thus if MinimumGain = 2, once you have typed "we" then "were" can be predicted, but "web" cannot.
The default value is 0 which includes all words in the available data.
If both MinimumLength and MinimumGain are specified then both are used - only words which fit both conditions are included.

Case = *Lower* or *Normal* or *InitialLetter* or *Upper*
Controls whether the results are displayed in upper or lower case. The default is Normal, in which case words are displayed as they are stored in the data. I.e. words which have a capital letter are displayed as such.
The possible options are:

- **Lower**: text is entirely in lower case, even if the word would normally have a capital letter, or be all capitals, e.g. "hello stuart"
- **Normal**: words are shown as they are stored, i.e. usually in lower case, e.g. "hello Stuart". If a word is typed with a capital, then all predictions are shown capitalised.
- **InitialLetter**: the first letter of every word is capitalised, e.g. "Hello Stuart". Words in all capitals in the lexicon (BBC, USA) remain in all capitals, and are not converted.
- **NextInitialLetter**: as above for the next word only. Then the state is automatically reset to *Normal*.
- **Upper**: the text is entirely in capitals, e.g. "HELLO STUART"

PunctSpace = *Off* or *Single* or *Double*
This controls whether spaces are added automatically after every punctuation mark. If Off automatic spaces are only added after words selected from the predicted words. Single adds one space after any punctuation mark. Double adds two spaces after '.' and '?', but still a single space after other marks.
These automatic spaces are removed again if another punctuation mark immediately follows the first – ie typing '…' produces exactly that with one or two spaces on the end, not '. . . '
The default is Off.
Note that opening brackets - ([and { - are ignored in punctuation processing. They are neither followed by automatic spaces, nor remove any automatic spaces which preceeded them.

Language = *code*
The code should be either a full culture code, eg:
en-GB = UK English

en-US = US English
sv-SE = Swedish (Sweden).

(The full list is here: http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo%28VS.80%29.aspx)
Or, more usually, the code can be just the first 2 letters of this to specify the language.
Different user data is stored for each language, but not culture (ie changing from "en" to "sv" loads different user data files, but changing "en-GB" to "en-US" does not). The system data is dependent on the files installed with SAW. At the time of writing this is likely to be one generic file each for English ("en") and Swedish ("sv"). Therefore at the moment Blade doesn't distinguish between UK and US English, for example.
This defaults to the language of the computer.

There is one other setting described in the Blade technical document: *NumberPredictions*. Using this in SAW has no effect - SAW automatically sets the number of predictions to fill the available spaces in the selection set. If this setting is included, it is just ignored.

Script restrictions

The script item "wordlist set" should only be used in the start-up script of a selection set. If the selection set has more than one word prediction area (because of pop ups) it is possible to use this command up to a maximum of 4 times in the start-up script to initialise up to 4 prediction areas.
If this command is used more than once with the Blade prediction engine, the different prediction areas will use the same settings. All the settings specified in all of the "wordlist set" command or "blade settings" command will be applied to all the word predictions

DDE

Some of the older prediction engines could be configured and controlled using DDE commands. These do not apply to Blade – any DDE commands directed at Blade will be ignored (it is not a separate program, so cannot received DDE independently)

Data files

Blade has 3 data files. The first is the standard word frequency data for your language, which was generated by scanning a reasonably large amount of text (currently 15 million words in English). This file is installed by SAW.

There are also 2 files containing user data:

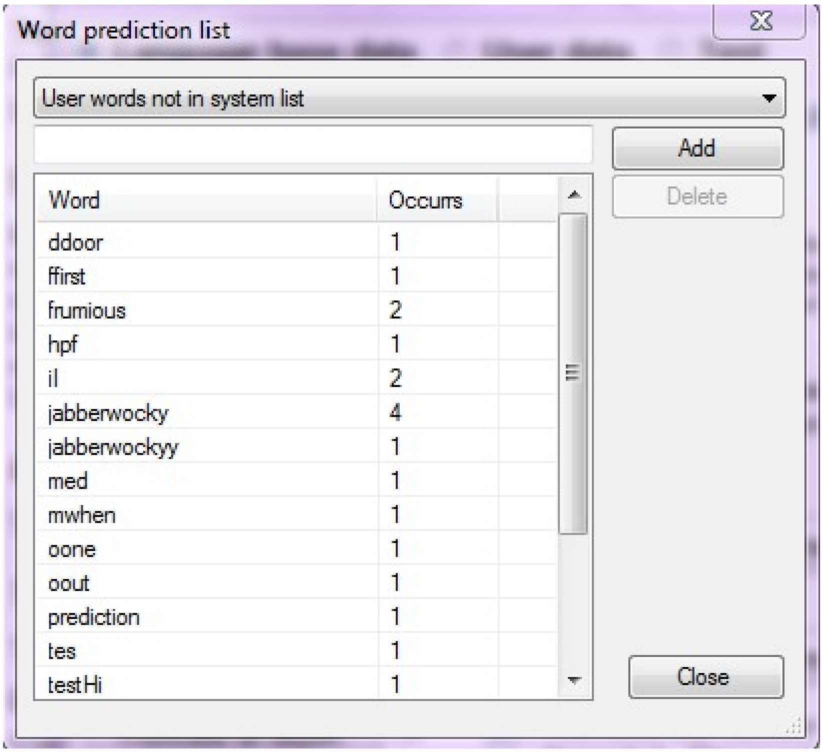
- Data-XX.bin (where XX is a 2-digit language code)
- UserData-XX.bin

These are automatically created in the "local application data" folder. On Windows 7 this is: "c:\Users\user_name\AppData\Local\Blade", but was different under XP.
If just using SAW on one machine you should not need to do anything with these files. However if you want to transfer the information Blade has learned about your use of language to another machine copy the UserData-XX.bin file across to the correct location on the other machine.
It isn't essential to copy "Data-XX.bin" which is likely to be a much larger file - this file is a combination of UserData and the language base data, and can be rebuilt automatically if it is missing.

Editing the word list

In SAW, there is an entry on the "View" menu: "View word prediction words", which views and edits the user words. Blade stores 2 word lists: the system vocab, installed with SAW, and the

words which the user has typed. These are combined to make the list from which predictions are drawn. Most of the changes are made in the user word list.
The list editor looks like this:



Word	Occurrs
ddoor	1
ffirst	1
frumious	2
hpf	1
il	2
jabberwocky	4
jabberwockyy	1
med	1
mwhen	1
oone	1
oout	1
prediction	1
tes	1
testHi	1

The initial view lists all the words which the user has typed, *which do not appear in the system vocab*. This is the best view if you want to clean up incorrect words which have been gathered by the prediction system.
The second column lists how many times the user has typed the word (this does not include the number of appearances of this word in the system vocab).
The drop down list at the top of the page can change to other views:

- All words which the user has typed (whether they appear in the system vocab or not)
- Words which have been deleted from the system vocab – see below.
- Abbreviations, i.e. user defined abbreviations and expanded text strings

To add new words, type them into the box above the list and click “Add”. Clicking Add more than once adds occurrences to the word, which will make it more likely to be predicted.

To delete a word, select it in the list, and click Delete.

- If the word appears only in the user list, and not the system vocab then it is simply removed. If it is typed again it will be added back to the user list.
- If it appears in the system list, then it is deleted completely and the software remembers that it has been deleted. The word will never be predicted, and will not reappear in the list if it is typed again. Also if the software is updated and a new version of the system vocab is installed, the word will not reappear. Ie Blade keeps a list of these deleted words to prevent them being added back accidentally during updates.
  If you do want to add the word back to the predictions, change the view to list the deleted words, select it, and click Add. This restores it to the prediction system, and all the system probability data for the word is restored (its frequency, and which words it commonly follows)

Changes made in this screen are automatically saved when SAW is closed. If you make a lot of changes, there may be a pause of a couple of seconds while Blade re-optimises the combined prediction data from the system vocab + user words.