Technieken

Op deze pagina vindt u een aantal problemen waarvoor Suares & Co oplossingen zocht.

Technology

On this page some solutions we found for some uncommon issues.

Creating a hunspell dictionary for use as spellcheck in Open Office

25-10-2008

Cool! OpenOffice/LibreOffice uses hunspell as spellchecker. This means that it is possible to create your own wordlist and affix file for your language, if that doesn't exist. Well, for papiamentu - Curaçao's native language - such dictionary doesn't exist. But I had a hard time finding out how to create such files, so I am documenting it here...

Installing hunspell on Ubuntu 8.04

Of course, you need hunspell installed. I am sure you had that already, and if not, the following might help:

```
sudo apt-get install hunspell hunspell-tools
```

The wordlist

A wordlist is just a list of words. It's the base for the dictionary. A wordlist can have 8000 words, or 100.000, or whatever seems reasonable for your language. Papiamentu has about 30.000 words at the moment.
Here's a very simple example wordlist:

```
love
lover
lovers
beer
beers
office
open
opened
chair
```

Save that lists as wordlist, just for now. Make sure there's a newline at the end of the file or else that last character will be eaten!

The Dictionary File

A dictionary file is just a wordlist preceded by the number of words. So this will make a dictionary file (sorted, too, and duplicates removed):

```
wc -l wordlist > yourlang.dic
sort wordlist | uniq >> yourlang.dic
```

How to create the Affix File

An affix file is ehmm... quite complex. It took me a couple of hours to understand that it can just be an empty file... if you have a relatively small wordlist (let's say less then 100.000).
Create an empty affix file:

```
touch yourlang.aff
```

In the Right Place

On Ubuntu 8.04, the dictionaries are kept in /usr/share/myspell/dicts/. So copy our language there:

```
sudo cp yourlang.* /usr/share/myspell/dicts/
```

Testing the Hunspell

```
hunspell -d yourlang
```

This should give you a prompt. Enter a word and you will see a result:

```
Hunspell 1.1.9
love
*
```

This means, that the word 'love' is spelled correctly according to your language.
Now try a non-exsiting word:

```
bove
& bove 1 0: love
```

This means that 'bove' is not a word, but 'love' comes close.

More Affixion

You got to read the manual on the affix file format. Here is a less comprehendable but more comprehensive one. Just a small example:

```
SET UTF-8

SFX P Y 1
SFX P 0 s
```

It says the rule 'P' adds an 's' behind a word without removing any characters. It's a totally bogus rule for the english language, but it'll work fine with our example dictionary.
I also added the SET UTF-8 because in papiamentu, my accents got garbled.

Munch the .dic and the .aff

Munching will apply the affix rules to the dictionary, and produce a smaller dictionary. In fact, it will replace 'lover' with 'lover/P' and remove 'lovers'. One word less. Because the rule will discover these two words and decide that it's more efficient to use 'lover/P'. It'll also remove 'beers' and replace 'beer' with 'beer/P'. Munching will also add the wordcount at the beginning of the .dic file.

```
munch yourlang.dic yourlang.aff > yournewlang.dic
mv yourlang.dic yourlang.dic.old
mv yournewlang.dic yourlang.dic
sudo cp yourlang.* /usr/share/myspell/dicts/
```

Now test it again:

```
hunspell -d yourlang
Hunspell 1.1.9
love
*
```

```
lover
*

lovers
+ lover
```

Ah! 'love' and 'lover' as expected, and 'lovers' has 'lover' as base.. that's what the '+' means. But remember, for small wordlists, the affix file can be empty!

For the spellchecker to guess the correct 'suggestions', a frequency list of characters is needed. You can produce one like this:

```
tr -d "\n" < words \
| while read -n1 char; \
do echo $char; \
done \
| sort | uniq -c| sort -rn
```

That will give you a list like this:
14177 a
11060 i
10470 e
10172 o
9771 n
8638 s

You can take it a step further:

```
tr -d "\n" < words \
| while read -n1 char; \
do echo $char; \
done \
| sort | uniq -c| sort -rn \
| sed "s/^.* //" \
|tr -d "\n"
```

This will give you, for example:
aieonstrkldmuphbgáfvèòcóíyéwñzjùúüABCSKIHEPLGTFxMJRqYXVDç