Hunspell

Данный документ является переводом официальной документации системы проверки орфографии Hunspell, который выполнили члены сообщества проекта Mozilla Россия:

Тимур Тимирханов,
Игорь Любимов,
Александр Словесник,
Алексей Губанов.

Содержание

Название

hunspell – формат словарей Hunspell и файлов аффиксов

Описание

Для проверки орфографии Hunspell требуется два файла. Первый файл - словарь, содержащий слова, второй - файл аффиксов, который определяет значения специальных меток (флагов) в словаре.

Файл словаря (.dic) содержит список слов, по одному слову в строке. В первой строке словарей (за исключением персональных словарей) указывается приблизительное количество слов в словаре (для оптимального распределения памяти). После каждого слова может следовать слэш ("/") и один или более флагов, соответствующих аффиксам и атрибутам. Слова в словаре также могут содержать слэши, экранированные "". По умолчанию, флаг представляет собой один (обычно, алфавитный) символ. В файле словаря Hunspell также может существовать поле для морфологического описания, отделяемое табуляцией.

Формат морфологического описания определяется пользователем.

Файл аффиксов (.aff) может содержать необязательные атрибуты. Например, **SET** для определения кодировки символов файлов аффиксов и словаря. **TRY** определяет заменяемые символы для предлагаемых замен. **REP** определяет таблицу замен для исправлений нескольких символов. **PFX** и **SFX** определяют классы префиксов и суффиксов, обозначенных флагами аффиксов.

Следующий образец файла аффиксов определяет кодировку символов UTF-8. Предлагаемые замены **TRY** отличаются от неправильного слова на одну букву или апостроф. С помощью этих флагов **REP**, Hunspell предлагает правильное слово, если вместо *f* напечатано *ph* или наоборот.

```
SET UTF-8
TRY esianrtolcdugmphbyfvkwzESIANRTOLCDUGMPHBYFVKWZ'

REP 2
REP f ph
REP ph f

PFX A Y 1
PFX A 0 re .

SFX B Y 2
SFX B 0 ed [^y]
SFX B y ied y
```

В этом файле определено 2 класса аффиксов. Класс A определяет префикс *re-*. Класс B — два суффикса *-ed*: один для слов, оканчивающихся не на *y* и второй — для оканчивающихся на *y*. (См. подробное описание ниже.) Эти классы аффиксов используются следующим файлом словаря.

```
3
hello
try/B
work/AB
```

В этом случае, правильными словами являются: *hello*, *try*, *tried*, *work*, *worked*, *rework*, *reworked*.

Общие флаги

SET *кодировка*

Устанавливает кодировку символов для слов и морфем в файлах словаря и аффиксов. Поддерживаемые значения: UTF-8, ISO8859−1 – ISO8859−10, ISO8859−13 – ISO8859−15, KOI8-R, KOI8-U, microsoft-cp1251, ISCII-DEVANAGARI.

FLAG *значение*

Устанавливает тип флагов. Тип флага по умолчанию — символ расширенного набора ASCII (8-битового). Значение *UTF-8* устанавливает символы Unicode в кодировке UTF-8. Значение *long* устанавливает в качестве флага два символа расширенного набора ASCII, *num* — десятичные числа от 1 до 65535, разделяемые запятыми. Баг: Тип флага UTF-8 не работает на платформе ARM.

COMPLEXPREFIXES

Включает разбор двойных префиксов и одинарных суффиксов для агглютинативных языков с системой письма справа налево.

LANG *код_языка*

Устанавливает код языка. В Hunspell могут содержаться участки кода, действующие для конкретного языка и включаемые флагом **LANG**. В настоящее время существуют участки кода, действующие для языков с кодами az_AZ, hu_HU, TR_tr (см. исходный код).

AF *количество_ссылок_на_флаги*
AF *вектор_флага*

Hunspell может заменять наборы флагов аффиксов численным значением (сжатие ссылок). Первый пример с включенным сжатием ссылок:

```
3
hello
try/1
work/2
```

Определения **AF** в файле аффиксов:

```
SET UTF-8
TRY esianrtolcdugmphbyfvkwzESIANRTOLCDUGMPHBYFVKWZ'
AF 2
AF A
AF AB
```

Также см. примеры в каталоге tests/alias*.

Примечания:
1. Если в файле аффиксов содержится параметр **FLAG**, поместите его до определений **AF**.
2. Используйте утилиту makealias в составе Hunspell для сжатия файлов .aff и .dic.

AM *количество_ссылок_на_морфологические_описания*
AM *морфологическое_описание*

Hunspell может также заменить морфологические описания в правилах аффиксов численным значением (сжатие ссылок). См. примеры в папке tests/alias*.

Флаги для предложений

TRY *символы*

Hunspell может предлагать правильные слова, отличающиеся от введённых на один символ в флаге **TRY**. Значения **TRY** чувствительны к регистру.

NOSUGGEST *флаг*

Слова, помеченные флагом NOSUGGEST, не предлагаются. Флаг можно использовать для грубых и непристойных слов.

MAXNGRAMSUGS *число*

Устанавливает число предложений, определяемых по вероятности последовательного появления символов. Значение 0 отключает такие предложения.

NOSPLITSUGS

Отключает предложения для слов, пишущихся через дефис.

SUGSWITHDOTS

Добавляет точку(и) к предложениям, если введенное слово оканчивается на точку(и). (Не требуется для словарей OpenOffice.org, так как в OpenOffice.org имеется алгоритм автоматической расстановки точек.)

REP *количество_определений_замены*
REP *заменяемое_замена*

В файлах аффиксов (.aff) с помощью таблицы замен определяется фонетическая информация для конкретного языка. Сначала идет **REP** с заголовком таблицы, а затем две и более строки **REP** с данными. С помощью этих таблиц Hunspell предлагает правильные варианты написания для слов, если правильное написание слова отличается от вводимого более чем на 1 символ. Например, таблица замен для английского языка, которая обрабатывает неправильно написанные согласные:

```
REP 8
REP f ph
REP ph f
REP f gh
REP gh f
REP j dg
REP dg j
REP k ch
REP ch k
```

Примечания:
1. С помощью таблицы **REP** можно определять замены для наиболее типичных опечаток, отличающихся на 1 знак. У этих замен приоритет выше, чем у предложений **TRY**.
2. Таблица замен также может быть использована для более жесткой проверки сложносоставных слов (запрет на сгенерированные сложносоставные слова, если они являются одноосновными словами с обычными ошибками, см. **CHECKCOMPOUNDREP**).

MAP *количество_определений_в_таблице*
MAP *строка_связанных_символов*

С помощью таблицы связи символов для каждого конкретного языка в файле аффиксов (.aff) можно определить информацию о символах, связанных друг с другом больше, чем символы вне таблицы. С ее помощью Hunspell предлагает правильное написание для слов, при вводе которых неправильная буква из соотнесенного набора была введена более одного раза.

Например, для немецкого языка одним из возможных наборов связи является связь между *ü* (с умляутом) и *u* (без умляута). Слово *Frühstück* необходимо писать с двумя *u* с умляутом, а не без.

```
MAP 1
MAP uü
```

Флаги для образования сложносоставных слов

BREAK *количество_определений_разделений*
BREAK *символ_или_последовательность_символов*

Определяет точку для разделения слова и проверку полученных частей по отдельности. Полезно для сложносоставных слов, объединенных с помощью

одного символа или последовательности символов (например, дефис в английском и немецком или дефис и короткое тире в венгерском языках). Тире не рекомендуется использовать как точку для разбиения слов, т.к. сложносоставные слова с тире могут содержать неправильные части. Используя **BREAK**, Hunspell может проверить обе части сложносоставных слов, разделяя слова по обычным и коротким тире:

```
BREAK 2
BREAK -
BREAK -- # n-dash
```

Разделение рекурсивно, так что foo-bar, bar-foo и foo-foo--bar-bar будут являться правильными сложносоставными словами.

Примечание: Примечание: флаг **COMPOUNDRULE** лучше (или будет лучше) подходит для обработки тире и других символов и их сочетаний для образования сложносоставных слов. Используйте **BREAK**, если вы хотите проверять слова с тире или другими соединительными символами и у вас нет времени или возможности написать точные правила образования сложносоставных слов с использованием **COMPOUNDRULE** (**COMPOUNDRULE** пока может обрабатывать только последний суффикс составного слова).

Примечание II: Для проверки орфографии из командной строки, установите параметры **WORDCHARS**: *WORDCHARS -* (см. примеры tests/break.*)

COMPOUNDRULE *количество_шаблонов_словосложения*
COMPOUNDRULE *шаблон_словосложения*

Определяет шаблоны образования сложносоставных слов с использованием regex-подобного синтаксиса. Первое **COMPOUNDRULE** является заголовком, в котором указано число следующих за ним шаблонов **COMPOUNDRULE**. Шаблоны сложных слов состоят из флагов для образования сложносоставных слов и звездочки или знака вопроса, являющихся мета-символами. Флаг, за которым следует *, соответствует слову, включающему 0 и более слов, помеченных им. Флаг, за которым следует *?*, соответствует слову, включающему 0 или 1 слов, помеченных им. См. примеры compound*.* в каталоге tests.

Примечание: мета-символы * and *?* работают только со следующими типами **FLAG**: 8-битные (по умолчанию) и UTF-8.

Примечание II: Флаги **COMPOUNDRULE** не совместимы с флагами словосложения **COMPOUNDFLAG**, **COMPOUNDBEGIN**, и т.д. (используйте эти флаги для разных слов).

COMPOUNDMIN *число*

Минимальная длина слов в составных словах. Значение по умолчанию - 3 буквы.

COMPOUNDFLAG *флаг*

Слова, отмеченные **COMPOUNDFLAG**, могут являться частью составного слова (за исключением случаев, когда слово имеет длину меньшую, чем указано в COMPOUNDMIN). Аффиксы с COMPOUNDFLAG также допускают

использование слов с ними в составе сложных слов.

COMPOUNDBEGIN *флаг*

Слова, отмеченные COMPOUNDBEGIN (или с отмеченным аффиксом), могут быть первыми элементами составных слов.

COMPOUNDLAST *флаг*

Слова, отмеченные COMPOUNDLAST (или с отмеченным аффиксом), могут быть последними элементами составных слов.

COMPOUNDMIDDLE *флаг*

Слова, отмеченные **COMPOUNDMIDDLE** (или с отмеченным аффиксом), могут располагаться в середине составных слов.

ONLYINCOMPOUND *флаг*

Суффиксы, отмеченные **ONLYINCOMPOUND**, могут находиться только в составных словах (словосоединяющие морфемы в немецком и шведском языках). Флагом **ONLYINCOMPOUND** также можно отметить обычные слова (см. примеры в tests/onlyincompound.*).

COMPOUNDPERMITFLAG *флаг*

По умолчанию, префиксы могут располагаться только в начале составных слов, суффиксы могут располагаться только в конце составных слов. Аффиксы, отмеченные флагом **COMPOUNDPERMITFLAG**, могут располагаться и внутри составных слов.

COMPOUNDFORBIDFLAG *флаг*

Суффиксы с этим флагом запрещают использование слов с данным аффиксом в составе сложносоставных.

COMPOUNDROOT *флаг*

Флаг COMPOUNDROOT помечает сложносоставные слова в словаре (Сейчас он используется только в коде для венгерского языка).

COMPOUNDWORDMAX *число*

Устанавливает максимальное количество слов в составном слове. (По умолчанию, оно не ограничено.)

CHECKCOMPOUNDDUP

Запрещает повторение слов в составных словах (например, foofoo).

CHECKCOMPOUNDREP

Запрещает образование составного слова, если данное (обычно неправильное) составное слово может являться обычным словом с ошибкой, определяемой флагом **REP**. Полезно для языков, склонных к образованию составных слов.

CHECKCOMPOUNDCASE

Запрещает начинать слова в составных словах с прописной буквы.

CHECKCOMPOUNDTRIPLE

Запрещает образование составного слова, если в нем будет идти три одинаковые буквы подряд (например, foo|ox или xo|oof). Баг: отсутствует поддержка мультибайтовых символов для кодировки UTF-8 (работает только для 7-битных ASCII символов).

CHECKCOMPOUNDPATTERN *число_определений_checkcompoundpattern*
CHECKCOMPOUNDPATTERN *конечные_символы_начальные_символы*

Запрещает образование составных слов, если первое слово в нем заканчивается на конечные_символы, а следующее слово начинается на начальные_символы.

COMPOUNDSYLLABLE *максимальное_число_слогов гласные*

Требуется для специальных правил образования составных слов в венгерском языке. Первый параметр — максимальное число слогов, которые могут иметься в составном слове, если количество слов в составном слове больше, чем **COMPOUNDWORDMAX**. Второй параметр — список гласных (для вычисления слогов).

SYLLABLENUM *флаги*

Нужен для специальных правил образования составных слов в венгерском языке.

Флаги для создания аффиксов

PFX *название_аффикса разрешено_соединение число_правил*
PFX *название_аффикса отсекаемые_символы условие морфологическое_описание*
SFX *название_аффикса разрешено_соединение число_правил*
SFX *название_аффикса отсекаемые_символы условие морфологическое_описание*

Аффикс может быть как префиксом, так и суффиксом, которые присоединяются к корням слов для образования новых. Возможно определять классы аффиксов с произвольным числом правил аффиксации. Классы аффиксов объявляются с помощью флагов для создания аффиксов. Первая строка в определении класса аффиксов является заголовком, в котором могут быть следующие поля:

(0) Название флага (**PFX** или **SFX**);
(1) Название класса аффикса;
(2) Разрешение на соединение префиксов и суффиксов. Возможные значения: Y (да) или N (нет);
(3) Число строк с правилами.

Поля правил аффиксации:

(0) Название флага;
(1) Название аффикса;
(2) Отсекаемые символы из начала (если аффикс — это префикс) или конца (если аффикс — это суффикс) слова;
(3) Аффикс (затем - опциональные названия классов, которые могут продолжить слово, разделяемые слэшами);
(4) Условие;

Нулевой аффикс или отсечение обозначаются нулями, нулевое условие — точкой. Условие — это простое правило, построенное по принципу

регулярных выражений. Условие необходимо выполнить до применения аффикса к слову. (Точка обозначает любой символ. Символы в квадратных скобках обозначают любой символ из заданного набора символов. Крышечка (^) вблизи первой фигруной скобки определяет дополнительный набор символов. Тире ничего не обозначает.)

(5) Морфологическое описание производного формата.

Прочие флаги

CIRCUMFIX *флаг*

Суффиксы с флагом CIRCUMFIX могут быть в слове, только если в этом слове есть префикс с флагом **CIRCUMFIX**, и наоборот.

FORBIDDENWORD *флаг*

Этот флаг обозначает запрещенную форму слова. Так как запрет влияет также на формы с аффиксами, то все формы этого слова: с аффиксами и в составе сложносоставных слов — также запрещаются.

KEEPCASE *флаг*

Для слов, отмеченных флагом **KEEPCASE**, запрещается образование форм в верхнем регистре и форм, начинающихся с прописной буквы. Полезно для специальных орфографий (в которых единицы измерений и символы валют не изменяют регистр даже в текстах, состоящих из прописных букв) и систем написания (например, для сохранения нижнего регистра символов IPA).

> **Примечание:** С декларацией **CHECKSHARPS**, слова с эсцетом и флагом **KEEPCASE** могут быть капитализированными и в верхнем регистре, но формы в верхнем регистре этих слов не могут содержать эсцет, а только *SS*. (См. пример germancompounding в каталоге tests дистрибутива Hunspell.)

LEMMA_PRESENT *флаг*

Обычно морфологический анализ выводит слова в словаре как глоссы. Иногда слова в словаре являются не глоссами, а основами с аффиксами и виртуальными основами. В этих случаях в морфологическое описание необходимо добавить глоссы (реальные основы). После добавления флага **LEMMA_PRESENT** к словам в словаре, вывод глосс, не являющихся настоящими, будет запрещен.

NEEDAFFIX *флаг*

Этот флаг отмечает в словаре виртуальные основы. Hunspell будет считать правильными только сочетания этих основ и аффиксов, за исключением случаев, если у этого слова есть омоним или нулевой аффикс. **NEEDAFFIX** также работает с префиксами и комбинациями префикс + суффикс (см примеры pseudoroot5.* в каталоге tests).

PSEUDOROOT *флаг*

Не поддерживается. (Бывшее имя флага **NEEDAFFIX**.)

WORDCHARS *символы*

WORDCHARS расширяет разметчик ввода командной строки Hunspell дополнительными символами для слов. Например, в венгерском языке

точка, обычное и короткое тире, числа и знак процента входят в состав слова.

CHECKSHARPS

Пара букв *SS* в немецких словах в верхнем регистре может являться эсцетом (*ß*) в верхнем реестре. Hunspell работает с этими формами с помощью флага **CHECKSHARPS** (см. также флаг **KEEPCASE** и примеры tests/germancompounding) при проверке правописания и предложении замен.

Морфологический анализ

Правила составления аффиксов Hunspell имеют необязательное поле для описания морфологии. Подобное поле, отделенное знаком табуляции, имеется и в файле словаря:

```
word/flags      morphology
```

Давайте определим простое слово с морфологической информацией. Файл аффиксов:

```
SFX X Y 1
SFX X 0 able . +ABLE
```

Файл словаря:

```
drink/X   [VERB]
```

Тестовый файл:

```
drink
drinkable
```

Тест:

```
$ hunmorph test.aff test.dic test.txt
drink:     drink[VERB]
drinkable: drink[VERB]+ABLE
```

Как видно из примера, анализатор соединяет поля морфологического описания в стиле item and arrangement.

Выделение двух суффиксов

Первоначально, алгоритм Ispell позволял выделять только один суффикс. Hunspell может выделять два суффикса.

Выделение двух суффиксов существенно улучшает обработку громадного количеством суффиксом, что является характерной особенностью агглютинативных языков.

Давайте расширим предыдущий пример, добавив второй суффикс (класс Y аффикса продолжает класс суффикса *able*):

```
SFX Y Y 1
SFX Y 0 s . +PLUR

SFX X Y 1
SFX X 0 able/Y . +ABLE
```

Файл словаря:

```
drink/X   [VERB]
```

Тестовый файл:

```
drink
drinkable
drinkables
```

Тест:

```
$ hunmorph test.aff test.dic test.txt
drink:      drink[VERB]
drinkable:  drink[VERB]+ABLE
drinkables: drink[VERB]+ABLE+PLUR
```

Теоретически при выделении двух суффиксов необходим квадратный корень(n) правил, где n -- число правил для суффиксов. Мы создали словарь проверки для венгерского языка с использованием выделения двух суффиксов.

Примечание: В грамматике препроцессора Hunlex можно выделять более двух суффиксов.

Расширенные классы аффиксов

Hunspell может обрабатывать более чем 65000 классов аффиксов. Вы можете использовать два новых синтаксиса для назначения флагов в файлах аффиксов и словаря.

Команда **FLAG** long устанавливает 2-х символьные флаги:

```
FLAG long
SFX Y1 Y 1
SFX Y1 0 s 1
```

Запись в словаре с флагами Y1, Z3, F?:

```
foo/Y1Z3F?
```

Команда **FLAG** num устанавливает цифровые флаги, разделяемые запятыми:

```
FLAG num
SFX 65000 Y 1
SFX 65000 0 s 1
```

Пример в словаре:

```
foo/65000,12,2756
```

Омонимы

В словаре Hunspell могут содержаться повторяющиеся слова-омонимы:

```
work/A    [VERB]
work/B    [NOUN]
```

Файл аффикса:

```
SFX A Y 1
SFX A 0 s . +SG3

SFX B Y 1
SFX B 0 s . +PLUR
```

Тестовый файл:

```
works
```

Тест:

```
> works
work[VERB]+SG3
work[NOUN]+PLUR
```

Эта возможность также позволяет запрещать некорректные комбинации префиксов/суффиксов в сложных случаях.

Зависимости префикс—суффикс

Интересным побочным эффектом многошагового выделения аффиксов является возможность корректной обработки циркумфиксов.

Например, в венгерском языке, превосходная степень прилагательных образуется одновременным присоединением префикса *leg-* и суффикса *-bb*. Одношаговое выделение не работает, так как в нем отсутствуют способы разрешения взаимосвязи между определенными префиксами и суффиксами, следовательно, неправильные формы считались правильными, например, **legvén = leg + vén* 'старый'. До введения кластеров, специальная обработка превосходных степеней была жестко зашита в код ранних версий HunSpell. Это могло быть разумно для одного случая, но зависимости префикс-суффикс встречаются везде при образовании слов с аффиксами, изменяющими их грамматические категории (сравни *payable*, *non-payable*, но не *non-pay* или *drinkable*, *undrinkable*, но не *undrink* в английском языке). Проще говоря, префикс *un-* возможен, только если основа *drink* имеет суффикс *-able*. При обработке этих шаблонов онлайновыми правилами аффиксов, а правила аффиксов проверяются по основе, отсутствует способ выражения такой зависимости, что неизбежно приводит к недостаточному или избыточному генерированию слов системой.

В следующем примере, класс суффикса R продолжается классом префикса P.

```
PFX P Y 1
```

```
PFX P   0 un . [prefix_un]+

SFX S Y 1
SFX S   0 s . +PL

SFX Q Y 1
SFX Q   0 s . +3SGV

SFX R Y 1
SFX R   0 able/PS . +DER_V_ADJ_ABLE
```

Словарь:

```
2
drink/RQ        [verb]
drink/S         [noun]
```

Морфологический анализ:

```
> drink
drink[verb]
drink[noun]
> drinks
drink[verb]+3SGV
drink[noun]+PL
> drinkable
drink[verb]+DER_V_ADJ_ABLE
> drinkables
drink[verb]+DER_V_ADJ_ABLE+PL
> undrinkable
[prefix_un]+drink[verb]+DER_V_ADJ_ABLE
> undrinkables
[prefix_un]+drink[verb]+DER_V_ADJ_ABLE+PL
> undrink
Unknown word.
> undrinks
Unknown word.
```

Циркумфиксы

Выделение условного аффикса при помощи продолжающих классов недостаточно для выражения циркумфиксов, так как они морфологически являются одним аффиксом. Для правильного морфологического анализа необходим флаг **CIRCUMFIX**.

```
# circumfixes: ~ obligate prefix/suffix combinations
# superlative in Hungarian: leg- (prefix) AND -bb (suffix)
# nagy, nagyobb, legnagyobb, legeslegnagyobb
# (great, greater, greatest, most greatest)

CIRCUMFIX X

PFX A Y 1
PFX A 0 leg/X .

PFX B Y 1
PFX B 0 legesleg/X .

SFX C Y 3
SFX C 0 obb . +COMPARATIVE
SFX C 0 obb/AX . +SUPERLATIVE
SFX C 0 obb/BX . +SUPERSUPERLATIVE
```

Словарь:

```
1
```

```
nagy/C  [MN]
```

Анализ:

```
> nagy
nagy[MN]
> nagyobb
nagy[MN]+COMPARATIVE
> legnagyobb
nagy[MN]+SUPERLATIVE
> legeslegnagyobb
nagy[MN]+SUPERSUPERLATIVE
```

Составные слова

За разрешение свободного образования составных слов приходится расплачиваться уменьшением точности распознавания, а также усложнением морфологического поиска и анализа. Хотя для разрешения образования составных слов из основ в Ispell были введены лексические переключатели, этого не достаточно. Например:

```
# affix file
COMPOUNDFLAG X

2
foo/X
bar/X
```

В этой записи как *foobar*, так и *barfoo* являются правильными словами.

Введение алгоритмов образования составных слов, чувствительных к направлению образования, т. е. появления возможности лексически явно указывать, может ли основа появляться как самая левая или самая правая компонента в составных словах, улучшило ситуацию. Тем не менее, этого всё же недостаточно для обработки сложных шаблонов составления слов, не говоря уже об уникальных для каждого языка норм расстановки переносов.

В настоящее время алгоритм Hunspell позволяет образование составных слов из любых форм слов, если они лексически помечены как потенциальные компоненты составного слова. В Hunspell этот алгоритм улучшен. Рекурсивные правила проверки составных слов делают возможной проверку сложных составных слов венгерского языка. Например, с помощью флагов **COMPOUNDWORDMAX**, **COMPOUNDSYLLABLE**, **COMPOUNDROOT** и **SYLLABLENUM** возможно записать правило образования составных слов венгерского языка *6—3*. Также, в венгерском языке, суффиксы часто изменяют возможности слова входить в состав составного. Hunspell позволяет помечать суффиксы флагами для образования составных слов. Имеются 2 специальных флага (**COMPOUNDPERMITFLAG** и **COMPOUNDFORBIDFLAG**), которые разрешают и запрещают использование производных в составе составных слов.

В Hunspell также имеется несколько возможностей для обработки составных слов немецкого языка:

```
# German compounding

# set language to handle special casing of German sharp s

LANG de_DE

# compound flags
```

```
COMPOUNDBEGIN U
COMPOUNDMIDDLE V
COMPOUNDEND W

# Prefixes are allowed at the beginning of compounds,
# suffixes are allowed at the end of compounds by default:
# (prefix)?(root)+(affix)?
# Affixes with COMPOUNDPERMITFLAG may be inside of compounds.
COMPOUNDPERMITFLAG P

# for German fogemorphemes (Fuge-element)
# Hint: ONLYINCOMPOUND is not required everywhere, but the
# checking will be a little faster with it.

ONLYINCOMPOUND X

# forbid uppercase characters at compound word bounds
CHECKCOMPOUNDCASE

# for handling Fuge-elements with dashes (Arbeits-)
# dash will be a special word

COMPOUNDMIN 1
WORDCHARS -

# compound settings and fogemorpheme for ‘Arbeit’

SFX A Y 3
SFX A 0 s/UPX .
SFX A 0 s/VPDX .
SFX A 0 0/WXD .

SFX B Y 2
SFX B 0 0/UPX .
SFX B 0 0/VWXDP .

# a suffix for ‘Computer’

SFX C Y 1
SFX C 0 n/WD .

# for forbid exceptions (*Arbeitsnehmer)

FORBIDDENWORD Z

# dash prefix for compounds with dash (Arbeits-Computer)

PFX - Y 1
PFX - 0 -/P .

# decapitalizing prefix
# circumfix for positioning in compounds

PFX D Y 29
PFX D A a/PX A
PFX D Ä ä/PX Ä
 .
 .
PFX D Y y/PX Y
PFX D Z z/PX Z
```

Образец словаря:

```
4
Arbeit/A-
Computer/BC-
-/W
Arbeitsnehmer/Z
```

Допустимые составные слова для словаря, представленного выше:

```
Computer
Computern
Arbeit
```

```
Arbeits-
Computerarbeit
Computerarbeits-
Arbeitscomputer
Arbeitscomputern
Computerarbeitscomputer
Computerarbeitscomputern
Arbeitscomputerarbeit
Computerarbeits-Computer
Computerarbeits-Computern
```

Недопустимые составные слова:

```
computer
arbeit
Arbeits
arbeits
ComputerArbeit
ComputerArbeits
Arbeitcomputer
ArbeitsComputer
Computerarbeitcomputer
ComputerArbeitcomputer
ComputerArbeitscomputer
Arbeitscomputerarbeits
Computerarbeits-computer
Arbeitsnehmer
```

Это решение неидеально и оно будет заменено алгоритмом проверки составления слов, основанным на шаблонах, который будет тесно интегрирован с разбором буфером ввода. Шаблоны, описывающие составные слова, представляют собой отдельные элементы ввода, связанные с высокоуровневыми свойствами их составляющих (например, число слогов, наличие аффиксов и дефисов). Для оценки правильности, эти шаблоны сравниваются с потенциальными отрывками составных слов.

Кодировка символов

Проблемы с восьмибитными кодировками

Ispell и Myspell используют восьмибитную кодировку ASCII , главным недостатком которой является неуниверсальность. Хотя венгерский язык имеет стандартную кодировку ASCII (ISO 8859-2), она не отражает отразить все нюансы орфографии. Например, в этом наборе символов отсутствует символ – (короткое тире), хотя он не только является официальным символов для выделения вводных предложений, но также может входить в состав слов как «длинный» дефис.

MySpell использует такие же восьмибитные таблицы символов, но в нём также есть и языки без стандартной восьмибитной кодировки. Например, множество африканских языков имеет нелатинские или расширенные латинские символы.

Использование оригинального написания некоторых иностранных имён, например, *Ångström* или *Molière* является нормой венгерской орфографии. А так как символы *Å* и *è* не входят в стандарт ISO 8859-2, то когда они используются вместе с символами, входящими только в стандарт ISO 8859-2 (например, окончания элатива *-bo=l*, аллатива *-to=l* делатива *-ro=l* с двойным акутом), получаются слова (например, *Ångströmro=l* или *Molière-to=l*), которые невозможно закодировать с помощью какой-либо одной кодировки символов ASCII.

Проблемы с восьмибитной ASCII кодировкой долго решались разработчиками Unicode. К сожалению, переход на Unicode (например, UTF-16) потребовал бы большой оптимизации кода и сказался бы на эффективности алгоритма. Алгоритм

Dömölki, используемый при проверке условий аффиксов использует 256-байтовые символьные массивы, которые разрослись бы до 64 КБ при кодировании в Unicode. Так как для интерактивной подстановки аффиксов для большого языка может потребоваться несколько сотен таких масивов (в случае венгерского языка количество масивов достигает 300 и даже больше, так как избыточное хранение одинаковых по структуре аффиксов повышает скорость алгоритма), переход на Unicode вызвал бы большие потери производительности. Тем не менее, ясно, что взамен потери производительности при независимости от кодировок, появятся огромные преимущества для многоязычных приложений, поэтому разработки в этом направлении долгое время были в наших планах.

Гибридное решение

Мы повысили эффективность при обработке Unicode. В не UTF-8 кодировках Hunspell работает с обычными восьмибитными кодировками, но со словарём и файлом аффиксов, кодированными в UTF-8, Hunspell использует гибридный тип работы со строками и проверкой аффиксов для поддержки Unicode:

Аффиксы и слова хранятся в UTF-8, во время анализа они, по большей части, обрабатываются в UTF-8, а при проверке условий и предложений конвертируются в UTF-16.

Алгоритм Dömölki используется для хранения и проверки семибитных ASCII таблиц символов (ISO 646), и сортирует списки в UTF-16 для символов Unicode для образцов условий.

На данный момент Hunspell поддерживает только первые 65536 символов (Основной Многоязычный Набор) из стандарта Unicode.