韓国語文章
処理시스팀에 관한 考察

柳　京　熙
〈KORSTIC 電算室長〉

1. 머리말

컴퓨터는 英数学를 쓰는 나라에서 만들어진 것이기 때문에 英数字의 처리에 있어 아주 效果的이다. 当初에는 컴퓨터가 数字처리를 위하여 만든 것이기는 하나 이것을 英文字処理에 応用하는 技術이 向上되어서 오늘날에는 文字처리가 一般化되었다. 그러나 文法이란 것은 数의 世界의 各種 法則보다 훨씬 어려우므로 文章処理技法이 確立되기 까지는 상당히 오래 걸린 셈이다.

데이터베이스(DB)에도 数値DB와 非数値DB가 있는데 非数値인 경우에는 그 実用化가 늦었기 때문에 数値DB만이 데이터 베이스를 代表하고 있는 것처럼 알려지고 있다. 그러나 데이터베이스의 效用性은 데이터의 共同利用과 情報検索에 있다고 볼 수 있기 때문에 数値DB만으로는 満足할 만한 情報検索을 期할 수가 없다. 特定한 事物에 관한 情報를 찾으려 할 때 그 事物의 分類記号나 符号를 일일히 외워두기가 힘들기 때문에 一般化하기 어려우며 오히려 말로서 찾는 것만 못한 경우도 많다. 따라서 情報検索에서 制限된 能力을 가지고 있는 数値DB보다 文章処理가 될 수 있는 DB가 바람직하며 경우에 따라서는 둘다 할 수 있는 것이 좋다.

筆者는 우리말 文章을 処理할 수 있는 " 우리말　DB시스팀"을 構想함에 있어서 尚存하고 있는 몇가지 問題点을 들어서 이에 관한 解決方案을 생각해 보겠다.

2. 우리글의 制限

한글専用과 漢字混用의 問題는 오래묵은 論争의 対象이며 앞으로도 언제 해결될지 모르는 커다란 課題이다. 한글의 字素와 모아쓴 字種에 관한 問題는 이미 解決된 것으로 보아도 좋을 것이다. 그러나 漢字의 使用制限問題는 여전히 남아 있다.

요컨데 常用漢字만으로 하느냐 拡張한다면 어떤 基準으로 하느냐 하는 것이다. 특히 人名, 地名과 같은 固有名詞로 된 漢字語때문에 字種의 数가 거의 無制限이 되어버린다. 筆者는 使用漢字를 確定지은 다음에 機械化한다는 것은 너무 長期間이 所要되기 때문에 使用頻度에 따라 第一水準 3,200字 第二水準 3,200字…… 등으로 確定해 나가면서 機械化를 서둘러야 할 것으로 생각된다.

3. 情報의 最小単位

컴퓨터가 文章処理를 함에 있어서 가장 基本이 되는 것은 字素와 字種別로 処理하거나 간에 同一한 結果를 얻는 것이 바람직하지만 이것이 必須要件이라고 하기는 어렵다. 漢字인 경우에는 字素(部首)別과 字種別 処理를 区分할 理由를 찾기가 어렵다. 왜냐하면 漢字의 字素(部首)의 数와 順序가 많고 複雑하기 때문에 한글, 알

表 1. 漢字音

가각간갈감값강개객갱각거건걸검겁게격견결겸겹
경계고곡곤골공과곽관괄광괘괴괵굉교구국군굴궁
권궐궤귀귁규균귤극근글금급긍기긴길김끽나낙난
날남납낭내냉냑넉년념넙넝노녹논농뇌뇨누눈눌
뉴뉵늑늠능니닉닌닐님다단달담답당대댁덕도독돈
돌동두둔득등라락란랄람랍랑래랭략량러력련렬렴
렵령례로록론롱뢰료룡루류륙륜률륭륵름릉리린림
립마막민말망매맥맹멱면멸명메모목몰몽묘무묵문
물미민밀비박반발방배백뱅번벌범법벽변별병보복
본봉부북분불붕비빈빙사삭산살삼삽상새색생서석
선설섬섭성세소속손솔송쇄쇠수숙순술숭쉬슬습승
시식신실심십쌍씨아악안알암압앙애액앵야약양어
억언얼엄업에여역연열염엽영예오육온올옹와완왈
왕왜외요욕용우욱운울웅원월위유육윤율융은을음
읍응의이익인일임입잉자작잔잠잡장재쟁저적건절
점접정제조족존졸종좌죄죠주죽준줄중즉즐즘즙증
지직진질짐집징차착찬찰참창채책처척천철첨첩청
체초촉촌총촬쵀최추축춘출충췌취측츤층치칙친칠
침칩칭타탁탄달탐답탕태택탱터토톤통퇴투특파판
팔패팽페편폄펑폐포폭표푼품풍피픽필핍하학한할
함합항해행향허헌헐험혁현혈험협형혜호혹혼홀홍
화확환활황홰회획횡효후훈훙흰훼휘휴휼흉흑흔흘
흠흡흥희히힐 合計 490字

파벳 및 数字와 같이 一般化하기 어렵기 때문이다. 따라서 文章処理를 위한 情報単位中 最小単位로서 字種이 될 수 밖에 없다. 그러나 한글에 限하여서는 最小単位가 字素로 될 수 있으나 文章処理에는 큰 意味가 없다.

컴퓨터의 設計에서 最小単位는 물론 1 Bit(비트)이나 8비트를 1 Byte(바이트)로 한 機械와 16비트以上을 1 Word(워드)로 한 機械가 있다. 1바이트를 単位로 하는 機械에서는 1 바이트에서

$$2^8=256$$

즉 256가지 字素나 字種밖에 表現할 수가 없다. 그러므로 2바이트에서 또는 16비트 1워드에서는

$$2^{16}=2^8\times2^8=256\times256=65,536$$

즉 65,536가지 字素나 字種을 表現할 수 있다.

어차피 한글만의 字種処理를 위하여서는 1 바이트 単位로 不可能하기 때문에 2바이트 単位가 되어야 한다. 2바이트 単位로 본다면 거의 모든 漢字의 字種도 커버할 수 있다.

4. 한글漢字의 入力方式

打字機나 텔렉스처럼 건반을 보지 않고 텍스트만 보면서 두손으로 찍는 方式을 터치(Touch) 方式이라 하고, 国漢文孔版打字機처럼 텍스트도 보고 건반도 보면서 찍는 方式을 사이트(Sight) 方式이라고 한다. 터치方式은 入力速度가 사이트方式에 比하여 훨씬(3~5倍) 빠르지만 取扱字種(또는 字素)이 数百에 不過하다. 数千의 字種을 入力할 必要가 있을 때에는 사이트方式밖에 없는데 万이 넘는 字種을 다룰 때에는 사이트方式도 問題가 있다.

現在 KORSTIC은 「KIPS-3」란 이름으로 한글—漢字自動変換시스템을 開発하고 있는데 이것은 漢字의 소리나는대로 한글의 字素를 터치方式으로 入力하여 컴퓨터로 하여금 ① 모아쓰기 処理 ② 낱말의 識別 ③ 該当 漢字말의 検索 ④ 漢字코우드로의 変換 ⑤ 漢字出力 등을 遂行하도록 하는 시스템을 말한다. 同音異語가 있는 경우에는 頻度順으로 테이블을 만들어 該當漢字말을 One-touch(원터치)方式으로 골라내도록 하였다. 어떠한 入力텍스트라도 어차피 手作業으로 校正하여야 하기 때문에 校正過程에서 다루도록 하면 될 것으로 보고 있다. 한편 筆者가 따로 調査한 바에 의하면 모든 漢字의 소리는 모두 490種에 不過하며, 가령 使用漢字를 10,000字로 定하더라도 한글 1字種当 약 520字의 漢字를 다룰 수 있다(표 1参照).

5. 文章処理의 要件

5.1 字素의 識別

이미 言及한대로 漢字의 字素(部首)의 識別은 特別한 意味가 없는 것 같다. 다만 한글의 字素識別은 모아쓰기 프로그램에 의한 処理過程에서 子母의 区分, 初中終声의 区分 등이 自動化되는 것이 바람직하다. 특히 한글入力의 効率化를 위하여 모아쓴 한글 字種을 破字統計하는 것도 必

要한 일이다.

5.2 字種의 識別

前述한대로 字種은 情報의 最小單位로 삼는것이 좋을 듯하다. 다만 한글인 경우에는 字素가 最小單位이기는 해도 "모아쓰기 프로그램"에 의해 字種으로 바꾸는 것이 바람직하다. 다만 한글字素데이티의 処理와 한글字種데이터의 処理에서 同一하게 結果를 내어야 한다는 固定観念은 없어도 좋을 것이다. 字種의 処理에서는 目的에 따라서 다르게 設計할 수 있으나 여기에서는 한글과 漢字의 区分이 自動的으로 되도록 하는 것으로 充分하다고 본다.

5.3 낱말의 識別

英文의 Text Processing을 위한 単語識別은 빈칸으로 하고 있다. 英文은 每単語마다 띄이쓰도록 되어 있기 때문에 単語識別을 自動化하기 쉽다. 그러나 우리말인 경우는 토씨는 붙여서 쓰도록 되어 있어서 英文과 같이 完全띄어쓰기가 아니라 半띄어쓰기이다. 이러한 경우의 単語識別을 위하여서는 데이터 自体의 加工이 必要하다. 즉 데이터를 完全띄어쓰기로 바꾸거나 또는 每토씨앞에 識別子를 追加人力토록 한다면 自動化가 可能하다. 筆者는 識別子의 機能으로서 슬래시(斜線)를 쓰는 것이 좋을 것이라고 본다. 컴퓨터가 빈칸, 슬래시 또는 기타 特殊文字로서 単語를 識別하고 出力시킬 때에는 슬래시를 無視해 버리면 된다.

5.4 不用語 또는 索引語의 識別

KWIC(Key Word in Context)이란 文章処理를 함에 있어서 重要語(Key Word)인지 아닌지를 識別하기 위하여 重要하지 않으면서 出現頻度가 높은 単語를 모아서 不用語(Stop Word) 目録을 컴퓨터에 담아두고, 識別된 모든 単語 가운데서 不用語가 아닌 単語를 重要語로 取扱하도록 만든 시스팀을 말한다.

이와 反対로 KWOC(Key Word out of Context)이란 重要한 単語를 미리 選定하여 컴퓨터에 담아두고 識別된 모든 単語가운데서 重要語와 一致하는 単語만 가려내는 시스팀을 말한다.

어느 것이나 重要度가 問題가 되며 重要度를 가리는 基準이 曖昧하다. KWIC인 경우에는 別로 重要하지 않은 単語가 重要語로 取扱될 우려가 있으며 KWOC인 경우의 重要度判定은 너무 主観的이다. 그러나 重要度判定을 위한 資料로서 KWIC부터 着手하는 것이 올바른 順序일 것이다.

KWIC시스팀인 경우에는 不用語인지 아닌지를 識別하게 되고 KWOC시스팀인 경우에는 重要度인지 아닌지를 識別하게 된다.

5.5 文章의 識別

英文이나 우리글의 文章은 句読点으로 区分하고 있다. 마찬가지로 컴퓨터도 句読点을 文章의 識別子로 삼으면 된다. 그러나 每文章마다 所在를 가리키는 別途의 識別子를 두는 것도 생각해 볼만한 일이다. 특히 데이터베이스를 만드는 경우에는 文章마다 꼬리표의 役割을 하는 識別子가 情報検索을 위하여 반드시 必要하다.

5.5 파라그래프의 識別

새로운 파라그라프를 쓸 때에는 一般的으로 줄을 바꾸고 몇칸의 빈칸을 둔 다음 시작한다. 시스팀도 물론 이것을 識別할 수 있어야 하며 追加해서 매 파라그래프마다 그 所在에 관한 情報, 그 파라그래프가 包含하고 있는 表現方法을 考慮해 볼만하다.

6. 데이터·코우드와 文章処理

情報処理能力을 極大化하기 위하여 意味를 가진 데이터를 略号나 数字로서 縮小表現하려는 試図는 오래전부터 있었다. 이에 따라 各 시스팀間의 데이터互換性을 向上시키기 위한 "데이터·코우드의 統一制定" 事業이 先進各国에서 相当히 推進되고 있으며 ISO의 여러가지 勧告도 있다.

우리나라에서도 内務部가 主管하여 制定 実施하고 있는 住民登録番号는 데이터 코우드의 役

割을 充分히 감당하고 있으며 이로써 個人의 識別을 하고 있다.

그밖에도 모든 意味를 가지는 事物의 데이터·코우드化는 必要하다.

그러나 데이터·코우드 制定을 위한 分類体系 確立의 어려움 그리고 分類体系変動에 따른 更新의 어려움 등으로 早速한 実現을 期하기는 어려운 実情이다.

情報検索이란 側面에서 볼 때 데이터·코우드에 의한 検索은, 시스팀은 索引語-데이터·코우드 対照表를 만든다든가 하는 追加的인 機能이 必要하고 利用者는 여기에 익숙하여야 한다든가 하는 要件이 수반된다. 대부분의 DBMS(데이터베이스 管理시스팀)에는 이른바 文章処理機能이 없다. 간혹 있는 것도 있으나 그나마 英語文章에서 単語程度에 不過하다.

理想的인 情報検索시스팀은 検索하고자 하는 実体의 全体를 모르더라도 皮相的인 프로필(편모)만 가지고도 実体를 알아낼 수 있어야 한다. 왜냐하면 情報要求者의 대부분은 必要한 情報를 分明하게 表現하지 못하기 때문이다. 코우드検索보다 重要語 또는 自然語에 의한 検索이 훨씬 利用하기 便利하다.

7. 맺음

文章処理技法은 모든 出版社와 新聞社와 같이 編輯校正을 必須的으로 遂行하여야 하는 곳의 電算化에는 반드시 考慮되어야 한다. 뿐만 아니라 索引作業에 莫大한 人力과 経費를 投入하고 있는 곳도 마찬가지이다. 또한 電算化된 資料를 순식간에 찾아 낼 수 있는 데이터·뱅크에는 더욱더 必要하다. 즉 編輯, 校正, 索引, 検索을 自動化할 수 있는 道具가 된다.

우리말 데이터베이스를 製作함에 있어서 컴퓨터 可読型으로 製作된 資料를 活用하면 製作費가 아주 싸진다. 데이터更新을 해야 할 경우에는 더욱더 값싸게 된다.

오늘날 우리가 必要로 하는 데이터의 管理가 너무 소홀한 傾向이 있다. 모든 데이터를 早速히 컴퓨터 可読型으로 바꾸는 作業에 着手하였으면 한다. 컴퓨터 시스팀, 데이터通信 등에 관한 問題는 데이터가 있고 나서부터 考慮되어도 늦지않다. 우리의 情報는 우리말로 入力処理, 出力될 수 있는 시스팀과 더불어 "National Data Base"가 構築되도록 関係当局에 建議하는 바이다.