دانشکده زبان‌ها و زبان‌شناسی

تمرینات درس

آشنایی با زبان‌شناسی رایانشی

استاد: دکتر محمد بحرانی

وحید مواجی
89702855

تابستان 1390

1. تحلیل‌گرهای مورفولوژی در بسیاری از سیستم‌های مرتبط با پردازش زبان طبیعی به کار می‌روند. در مورد نحوه کاربرد و مزایای به‌کارگیری تحلیل‌گرهای مورفولوژی در سیستم‌های مختلف پردازش زبان طبیعی بحث کنید.

کاربرد تحلیل‌گرهای صرفی در زبانشناسی رایانشی بسیار وسیع می‌باشد. از این برنامه‌ها می‌توان به عنوان برنامه‌های مستقل استفاده کرد یا بعنوان یک مرحله پیش‌پردازشی برای دیگر کارهای پردازش زبان طبیعی، مثلاً در Part Of Speech Tagging، پارسینگ و ترجمه ماشینی. نمونه‌ای از تحلیلگر صرفی زبان فارسی توسط نگارنده نوشته شده و در آدرس http://pars-morph.appspot.com موجود می‌باشد. از تحلیلگرهای صرفی می‌توان برای تحقیق روی صرف و نحو زبان در مطالعات محض زبانشناسی و آموزش زبان استفاده نمود. همچنین در کارهای پردازش زبان طبیعی نیز یک عنصر واجب به شمار می‌روند. مثلاً در مقابله‌گرهای املایی[1] مقایسه کلمه ورودی با یک لیست ثابت از کلمات غیرعملی و غیرممکن است چرا زبان ماهیت خلاقانه دارد و می‌شود کلمات بی‌شماری را با الگوهای یکسان ساخت. نمی‌شود به طور دستی همه گونه‌های تصریفی همه کلمات را لیست کرد. مثلاً آنطور که ما محاسبه کرده‌ایم برای فقط اسمهای فارسی، حداقل 200 گونه مختلف تصریفی متصور است. حال اگر تعداد اسم‌های فارسی را 50000 در نظر بگیریم (که تعداد واقعی خیلی بیشتر از این است)، حدود 10 میلیون ورودی باید در نظر بگیریم. با مقایسه عدد 50000 و 10 میلیون ارزش یک تحلیلگر صرفی در کم نگه‌داشتن اندازه حافظه مورد نیاز بیشتر معلوم می‌گردد.

ساختار صرفی یک کلمه که توسط یک تحلیلگر بدست می‌آید حاوی اطلاعات نحوی و گرامری آن کلمه نیز است، اطلاعاتی که برای Pos Tagging و پارس کردن جملات بسیار لازم است. به طور خاص این اطلاعات برای تجزیه وابستگی[2] بسیار مفید می‌باشد.

یک تجزیه‌گر صرفی مانند Pars-Morph خروجی‌اش شامل اجزا تشکیل دهنده کلمه و مقوله دستوری پایه و ریشه می‌باشد. این اطلاعات در ترجمه ماشینی بسیار مفید و ضروری می‌باشد چرا که حالت‌های مختلفی که برای معنی یک جمله متصور است بسیار به مقوله نحوی آنها ربط دارد. یعنی اگر نمودار تجزیه نحوی جمله مبدأ را داشته باشیم با مقوله‌های دستوری آنها، آنگاه این امر کمک می‌کند تا کلماتی از همان مقولات دستوری از زبان مقصد انتخاب شود و شباهت کلمات باعث ترجمه اشتباه نشود.

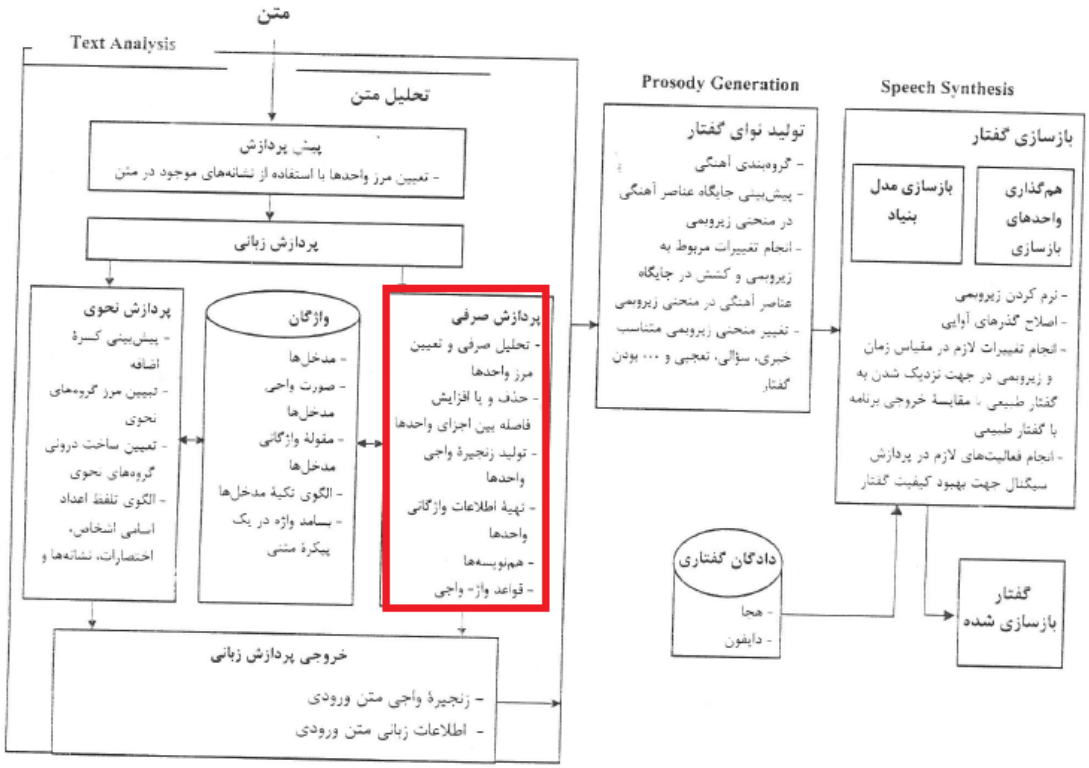
برای فرهنگ نویسی رایانه‌ای هم این مسأله ضروری است چرا که صورتهای تصریفی مختلف یک پایه یا ریشه را فقط به یک ورودی لینک می‌کنیم. در فرهنگ‌های کاغذی مشکلی که همیشه کاربران داشتند این بود که اگر شکل تصریفی یک کلمه را داشتند (مخصوصاً اگر بیقاعده بود) مثل went آنگاه معمولاً یافتن صورت اصلی کلمه بسیار سخت بود. ولی در فرهنگ‌های رایانه‌ای این مسأله حل شده و با دادن صورت تصریفی کلمه، معنی پایه یا ریشه آن می‌آید و می‌توانیم بفهمیم صورت تصریفی حاوی چه معنایی است.

یکی از بهترین نمودهای تحلیلگرهای صرفی در موتورهای جستجو مثل Google می‌باشد. وقتی کلمه‌ای را در موتور جستجو وارد می‌کنیم، احتمال خیلی زیادی وجود دارد که دیگر صورتهای تصریفی این کلمه نیز مرتبط با موضوع جستجو باشد. لذا بهترین کار این است که موتورهای جستجو، مطالب را بر اساس پایه کلمات ایندکس‌گذاری کرده و پرس و جوی ورودی

[1] Spell Checkers
[2] Dependency Parsing

کاربر را نیز وارد یک تحلیلگر صرفی کرده و از خروجی آن به عنوان ورودی پالایش شده موتور جستجو استفاده نمایند. این امر برای جستجو در مستندات گفتاری که موضوع سوال 4 نیز می‌باشد بسیار کاربرد دارد. چون مستندات گفتاری نیز باید با سیستم‌های بازشناسی گفتار به متن تبدیل شده و بر اساس خروجی یک تحلیلگر صرفی ایندکس شوند.

در سیستم‌های تبدیل متن به گفتار[3] نیز از تحلیلگرهای صرفی استفاده می‌شود. در این سیستم‌های برای بدست آوردن الگوی آوایی تلفظ باید ابتدا دانست پایه کلمه از چه نوع مقوله دستوری است و آهنگ کلمه و جمله را بر اساس الگوهای تکیه پایه و قوانین واج‌شناسی، بعد از چسبیدن وندها بدست آورد. نمودار یک سیستم تبدیل متن به گفتار و جایگاه تحلیل صرفی در آن در شکل زیر مشخص شده است.



نتیجه اینکه تحلیل‌گرهای صرفی کاربردهای فراوانی در بحث زبانشناسی و مهندسی زبان دارند که به طور خلاصه شامل موارد زیر می‌شوند:

- مطالعات زبان‌شناختی
  - ریشه‌یابی کلمات
  - جستجوهای گوناگون در پیکره‌های بزرگ متنی
  - تهیه فرهنگ‌های بسامدی، فرهنگ‌های طیفی، تهیه فهرست مدخل‌های انواع فرهنگ‌ها
  - دستور نویسی
- مطالعات مربوط به مهندسی زبان
  - تبدیل متن به گفتار

[3] Text To Speech

- ترجمه ماشینی
- استخراج اطلاعات از متون با حجم بالا
- خلاصه‌سازی متون
- طراحی موتورهای جستجوگر

2. گرامرهای مستقل از متن به تنهایی قادر به توصیف زبان طبیعی به صورت کامل و کارا نیستند. برای رفع این مشکل، گرامرهای محاسباتی جدیدی پیشنهاد شده‌اند که سعی می‌کنند نواقص گرامر مستقل از متن را بپوشانند. در مورد این گرامرها و نحوه پارس زبان طبیعی با آنها هر چه می‌دانید بنویسید.

در زبان‌های طبیعی معمولاً بین کلمات و عبارات تطابق وجود دارد. مثلاً گروه اسمی (یک مردان) یک جمله صحیح نیست چرا که (یک) نشاندهنده یک شی واحد است ولی (مردان) نشاندهنده یک شی جمع است. و گروه اسمی شرط شمار را رعایت نکرده است.

تطابق‌های دیگری هم وجود دارد مثل تطابق فعل-فاعل، جنس برای ضمایر، محدودیت هسته گروه و شکل متمم آن و غیره. برای حل این مشکل، فرمالیزم دستوری به گونه‌ای گسترش یافته است تا به سازه‌ها[4] امکان داشتن ویژگی بدهد. مثلاً می‌توانیم یک ویژگی NUMBER تعریف کنیم که دارای مقادیری s (برای مفرد) و p (برای جمع) باشد و سپس یک قانون CFG غنی‌شده[5] را به شکل زیر بنویسیم:

NP -> ART N only when $NUMBER_1$ agrees with $NUMBER_2$

این قاعده می‌گوید که یک NP درست از یک ART و یک N تشکیل شده است ولی فقط وقتی که ویژگی شمار کلمه اول با ویژگی شمار کلمه دوم بخواند. این یک قاعده با دو قاعده CFG برابر است که از نشانه‌های پایانی متفاوتی برای کدکردن شکل‌های مفرد و جمع همه گروه‌های اسمی استفاده می‌کنند مثل:

NP-SING -> ART-SING N-SING
NP-PLURAL -. ART-PLURAL N-PLURAL

درحالیکه این دو راهکار از لحاظ سادگی استفاده در این مثال یکسان به نظر می‌رسند ولی حالتی را درنظر بگیرید که همه قواعد گرامر که از یک NP در سمت راست خود استفاده می‌کنند باید تکرار شوند تا شامل قاعده‌ای شوند که دربرگیرنده NP-SING و NP-PLURAL باشد یعنی فی‌الواقع اندازه گرامر را دوبرابر می‌کنیم. کنترل ویژگی‌های دیگر مثل شخص اندازه گرامر را بزرگتر و بزرگتر می‌کند. با استفاده از ویژگی‌ها، اندازه گرامر تغییری نمی‌کند و محدودیت‌های تطابق هم رعایت می‌شود.
برای انجام این مهم، یک سازه به صورت یک ساختارویژگی[6] تعریف می‌شود – یک نگاشت از ویژگی‌ها به مقادیری که خصوصیات مرتبط یک سازه را تعریف می‌کنند. نام ویژگی‌ها در فرمول‌ها بصورت **برجسته** نوشته می‌شود. مثلاً یک ساختارویژگی برای سازه ART1 که نشاندهنده استفاده خاصی از کلمه است ممکن است به صورت زیر نوشته شود:
ART1: (CAT ART **ROOT** a **NUMBER** s**)**

[4] Constituents
[5] Augmented
[6] Feature Structue

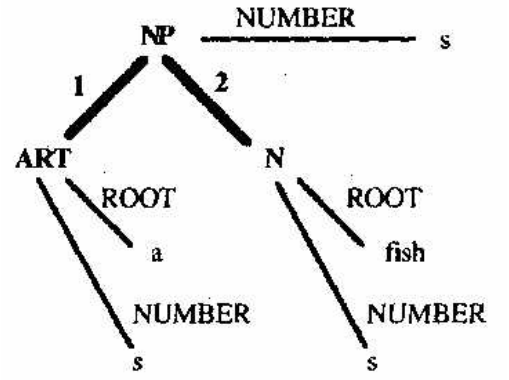
این فرمول بیان می‌دارد که این سازه‌ای است از مقوله ART که بعنوان ریشه شامل کلمه a است و مفرد می‌باشد. معمولاً یک اختصار استفاده می‌شود تا به مقدار CAT برجستگی بیشتری بدهد و شباهت بیشتری با CFG ساده داشته باشد. با استفاده از این اختصار، سازه ART1 بصورت زیر نوشته می‌شود:

ART1: (ART **ROOT** a **NUMBER** s**)**

از ساختارویژگی‌ها می‌توان برای نشان دادن سازه‌های بزرگتر هم استفاده نمود. برای انجام این کار خود ساختارویژگی‌ها می‌توانند بصورت مقادیر درآیند. ویژگی‌های خاصی که مبتنی بر اعداد صحیح 1و 2و 3 و غیره هستند برای اولین زیرسازه، دومین زیرسازه و غیره مورد نیاز می‌باشند. بدین ترتیب نمایش سازه NP برای عبارت "a fish" بدین شکل است:

NP1: (NP NUMBERs
1 (ART **ROOT** a **NUMBER** s**)**
2 (N ROOT fish **NUMBER** s**))**

باید توجه داشت که این را می‌توان بصورت نمایشی از یک درخت تجزیه نشان داد که زیرسازه‌های ویژگی‌های 1 و 2 متناظرند با لینک‌های زیرسازه‌ها در درخت.



قواعد یک گرامر غنی شده بصورت ساختارویژگی‌ها بیان می‌شوند نه بصورت مقوله‌های ساده. مقادیر ویژگی می‌توانند بصورت متغیرها بیان شوند به نحوی که یک قاعده را بتوان روی بازه وسیعی از حالات اعمال کرد. برای مثال یک قاعده برای NP های ساده می‌تواند به شکل زیر باشد:

(NP **NUMBER** ?n**) – (**ART **NUMBER** ?n**)**
(N **NUMBER** ?n**)**

این فرمول بیان می‌دارد که یک سازه NP می‌تواند از دو زیرسازه تشکیل شود؛ اولی یک ART باشد و دومی یک N که در آن ویژگی NUMBER در هر سه سازه یکسان است. بنابر این قاعده، سازه NP1 که قبلاً گفته شد یک سازه معتبر است. ولی از طرف دیگر سازه

* (NP **1** (ART **NUMBER** s)
2 (N **NUMBER** s))

تحت این قاعده معتبر نیست چرا که هیچ ویژگی NUMBER در NP وجود ندارد و سازه

* (NP NUMBER s
1 (ART NUMBER s)
2 (N NUMBER p))

بدین دلیل معتبر نیست که ویژگی NUMBER سازه N با دو ویژگی NUMBER دیگر یکسان نیست. متغیرها همچنین برای مشخص کردن ابهام در یک سازه بکار می‌آیند. مثلاً کلمه fish می‌تواند هم مفرد و هم جمع باشد و لذا ابهام دارد. لذا یک کلمه می‌تواند دو مدخل در واژگان داشته باشد که فقط با ویژگی NUMBER از هم تمیز داده می‌شوند. بالعکس می‌توانیم یک مدخل واخد تعریف کنیم که یک متغیر به عنوان مقدار NUMBER داشته باشد یعنی:
(N **ROOT** fish **NUMBER** ?n)

این امر امکان‌پذیر است چرا که مقدار ویژگی NUMBER برای کلمه fish مجاز است. در بسیاری از حالات، هرچند، نه تنها هر مقداری کار می‌کند بلکه بازه‌ای از مقادیر نیز ممکن است. برای مدیریت این حالت، متغیرهای محدود معرفی می‌شوند که متغیرهایی هستند که فقط یک مقدار را از یک لیست می‌پذیرند. مثلاً متغیر **?n{sp}** متغیری است که مقدار **s** یا مقدار **p** را می‌گیرد.

معمولاً وقتی چنین متغیرهایی را می‌نویسیم کلاً اسم متغیر را حذف می‌کنیم و فقط مقادیر ممکن را لیست می‌کنیم. با این فرض، کلمه fish را می‌توان به صورت سازه زیر نمایش داد:
(N **ROOT** fish **NUMBER** ?n{sp})
یا به شکل ساده‌تر:
(N **ROOT** fish **NUMBER** {sp})

مثال:

یک گرامر مستقل از متن غنی‌شده برای تولید زبان $L_n = \{a^n b^n c^n : n > 0\}$
در گرامر G_1 قواعد تولید L_n آورده شده است. نشانه‌های غیرپایانی که بیشتر از یکبار در تولید بکار می‌روند با اندیس‌هایی متمایز شده‌اند. نشانه غیرپایانی موجود در سمت چپ $\rightarrow$ اندیس صفر می‌گیرد مثلاً A_0 و نشانه‌های غیرپایانی تکرارشونده موجود در سمت راست فلش اندیس‌های 1و2 و ... از چپ به راست می‌گیرند. مثلاً A_1.COUNT یک ویژگی عددی هر کدام از نشان‌های غیرپایانی در G_1 است.

- $S \rightarrow A\ B\ C$, where A.COUNT := S.COUNT, B.COUNT := S.COUNT, C.COUNT := S.COUNT
- $A_0 \rightarrow a\ A_1$, where A_1.COUNT := A_0.COUNT - 1
- $B_0 \rightarrow b\ B_1$, where B_1.COUNT := B_0.COUNT - 1
- $C_0 \rightarrow c\ C_1$, where C_1.COUNT := C_0.COUNT - 1
- $A \rightarrow \varepsilon$, where A.COUNT = 0
- $B \rightarrow \varepsilon$, where B.COUNT = 0
- $C \rightarrow \varepsilon$, where C.COUNT = 0

یک گرامر مستقل از متن غنی‌شده برای پارس L_n
در اینجا گرامر G_2 آورده شده است که L_n را پارس می‌کند. یک اشتقاق با توجه به گرامر پارس مثل G_2 به طریق عکس گرامر تولیدکننده مثل G_1 ساخته می‌شود. این گرامر با یک رشته پایانی شروع می‌کند و هر خط از خط قبلی ساخته می‌شود با تعویض یک زیررشته با یک عنصر غیرپایانی و همه ویژگی‌هایش. اشتقاق تمام می‌شود اگر خط پایانی یک عنصر آغازین باشد. می‌توان دید که زبان پارس شده توسط G_2 همان زبان تولید شده توسط G_1 است. هر طبقه هم‌ارزی اشتقاق با توجه به پارسر را می‌توان به صورت یک درخت نشان داد. هرچند

توصیف ساختاری متناظر با اعداد L_n در G_2 متفاوت با G_1 است. برای تمایز قواعد گرامر پارسر از قواعد گرامر تولیدکننده، جهت فلش‌ها برعکس شده است.

- S ⟩ A B C, where A.COUNT = B.COUNT = C.COUNT
- A_0 ⟩ A_1 A_2, where A_0.COUNT := A_1.COUNT + A_2.COUNT
- B_0 ⟩ B_1 B_2, where B_0.COUNT := B_1.COUNT + B_2.COUNT
- C_0 ⟩ C_1 C_2, where C_0.COUNT := C_1.COUNT + C_2.COUNT
- A ⟩ a, where A.COUNT := 1
- B ⟩ b, where B.COUNT := 1
- C ⟩ c, where C.COUNT := 1

3. یکی از راه‌های رفع مشکل کمبود داده آموزشی در استخراج مدل‌های زبانی n-gram دسته بندی کلمات و استخراج مدل‌های n-gram مبتنی بر کلاس (class-based n-gram language models) می‌باشد. روش‌های مختلفی برای دسته‌بندی کلمات پیشنهاد شده است. روش‌های مختلف دسته‌بندی کلمات و نحوه استخراج مدل‌های n-gram مبتنی بر کلاس را به طور مشروح بیان کنید.

فرض کنید که واژگانی با V کلمه را به C کلاس تقسیم می‌کنیم با استفاده از تابع π که یک کلمه مانند w_iرا به کلاس c_i نسبت می‌دهد. می‌گوییم که یک مدل زبانی یک مدل n-gram مبتنی بر کلاس است اگر که یک مدل n-gram باشد و اگر علاوه بر آن برای
$1\le k\le n$ ، $Pr(w_k|w^{k-1}_1)=Pr(w_k|c_k)\ Pr(c_k|c^{k-1}_1)$

یک مدل n-gram مبتنی بر کلاس دارای $C^n-1+V-C$ پارامتر مستقل است. V-C تای آنها به شکل $Pr(w_i|c_i)$ هستند، به علاوه C^n-1 پارامتر مستقل از مدل زبانی n-gram برای واژگانی به اندازه C. لذا بجز حالت‌های خاصی که در آنها C=V یا n=1 است، یک مدل زبانی n-gram مبتنی بر کلاس همیشه پارامترهای مستقل کمتری نسبت به یک مدل زبانی n-gram کلی دارد.

با فرض داشتن متن آموزشی t^T_1، تقریب‌های درست‌نمایی بیشینه برای پارامترهای یک مدل 1-gram مبتنی بر کلاس به ترتیب زیر است:

$Pr(w|c) = C(w) / C(c)$

و

$Pr(c) = C(c) / T$

که منظور از C(c) تعداد کلمات موجود در t^T_1 است که کلاس آنها c می باشد. از این معادلات درمی‌یابیم که:

$c=\pi(w), Pr(w) = Pr(w|c)\ Pr(c) = C(w) / T$

برای یک مدل1-gram مبتنی بر کلاس، انتخاب نگاشت π تأثیری ندارد. برای یک مدل 2-gram مبتنی بر کلاس، تقریبهای درست‌نمایی بیشینه پارامترهای مرتبه دوم $Pr(t^T_2|t_1)$ را ماکزیموم می‌کند و به همین ترتیب $\log Pr(t^T_2|t_1)$ را و داریم:

$Pr(c_2|c_1) = C(c_1c_2) / \sum_c C(c_1c)$

طبق تعریف، $Pr(c_1c_2) = Pr(c_1)\ Pr(c_2|c_1)$ و لذا برای تقریب‌های درست‌نمایی بیشینه ترتیبی داریم:

$Pr(c_1c_2) = C(c_1c_2)/T \times C(c_1) / \sum_c C(c_1c)$

از آنجاییکه $C(c_1)$ و $\sum_c C(c_1c)$ تعداد کلماتی هستند که کلاس آنها به ترتیب در رشته‌های t^T_1و t^{T-1}_1، c_1 می باشد، عبارت پایانی در این معادله به سمت 1 میل می‌کند وقتی که T به سمت بی‌نهایت میل کند. لذا $Pr(c_1c_2)$ به سمت بسامد نسبی c_1c_2 میل می‌کند. فرض کنید:
$L(\pi) = (T-1)^{-1} \log Pr(t^T_2|t_1)$

آنگاه:

$$\begin{aligned} L(\pi) &= \sum_{w_1 w_2} \frac{C(w_1 w_2)}{T-1} \log \Pr(c_2 \mid c_1) \Pr(w_2 \mid c_2) \\ &= \sum_{c_1 c_2} \frac{C(c_1 c_2)}{T-1} \log \frac{\Pr(c_2 \mid c_1)}{\Pr(c_2)} + \sum_{w_2} \frac{\sum_w C(w w_2)}{T-1} \log \underbrace{\Pr(w_2 \mid c_2) \Pr(c_2)}_{\Pr(w_2)}. \end{aligned}$$

بنابراین از آنجاییکه $\sum_w C(ww_2)/(T-1)$ به سمت بسامد نسبی w_2 در متن آموزشی میل می‌کند و لذا به سمت $Pr(w_2)$ باید در حد داشته باشیم:

$$\begin{aligned} L(\pi) &= \sum_w \Pr(w) \log \Pr(w) + \sum_{c_1 c_2} \Pr(c_1 c_2) \log \frac{\Pr(c_2 \mid c_1)}{\Pr(c_2)} \\ &= -H(w) + I(c_1, c_2), \end{aligned}$$

که $H(w)$ آنتروپی توزیع کلمه 1-gram و $I(c_1,c_2)$ اطلاعات متقابل کلاس‌های مجاور می‌باشد. از آنجاییکه که $L(\pi)$ فقط از طریق این اطلاعت متقابل میانگین به π بستگی دارد، ناحیه‌ای که $L(\pi)$ را بیشینه می‌کند، در حد همان ناحیه‌ای است که اطلاعات متقابل میانگین کلاس‌های مجاور را بیشینه می‌کند.
هیچ روش عملی وجود ندارد که یکی از نواحی‌ای که اطلاعات متقابل میانگین را بیشنه می‌کند بدست آوریم. فی‌الواقع با داشتن چنین ناحیه‌ای، هیچ روش عملی وجود ندارد که نشان دهیم در حقیقت چنین امری ممکن است. هرچند با یک الگوریتم حریصانه می‌توان نتایج قابل قبولی کسب کرد. در ابتدا هر کلمه را به یک کلاس جدا نسبت می‌دهین و اطلاعات متقابل میانگین بین کلاس‌های مجاور را محاسبه می‌کنیم. سپس جفت کلاس‌هایی که میزان هدر رفتن اطلاعات میانگین متقابل در آنها کمترین است را با هم ادغام می‌کنیم. بعد از V-C تعداد از این ادغام‌ها، C کلاس باقی می‌ماند. اغلب برای کلاس‌هایی که با این روش بدست آمده‌اند، اطلاعات متقابل میانگین را می‌توان با انتقال برخی کلمات از کلاسی به کلاس دیگر افزایش داد. لذا بعد از استخراج مجموعه‌ای از کلاس‌ها از ادغام‌های متوالی، روی واژگان می‌چرخیم و هر کلمه را به کلاسی انتقال می‌دهیم که به ازای آن، ناحیه تشکیل شده بیشترین اطلاعات متقابل میانگین را داشته باشد. نهایتاً هیچ انتساب کلمه‌ای به اطلاعات بیشتری نمی‌انجامد. در این نقطه متوقف می‌شویم. ممکن است ناحیه‌ای بیابیم که اطلاعات متقابل میانگین بیشتری داشته باشد با انتساب همزمان دو یا بیشتر کلمه ولی چنین جستجویی آنقدر هزینه دارد که عملی نیست.

برای اینکه این الگوریتم زیربهینه را نیز حتی عملی سازیم باید توجه خاصی در پیاده‌سازی معطوف داریم. تقریباً $(V-i)^2/2$ ادغام داریم که باید برای انجام مرحله iام مدنظر قرار دهیم. اطلاعات متقابل میانگین باقیمانده بعد هر کدام از آنها، مجموع $(V-i)^2$ عبارت است که هر کدام یک لگاریتم دارند. از آنجاییکه کلاً باید V-C ادغام انجام دهیم، روش سرراست برای محاسبه از مرتبه V^5 است. به غیر از مقادیر بسیار کوچک V چنین محاسبه‌ای در تفکر نمی‌گنجد.

روش مقرون به صرفه دیگر باید از حشو موجود در این روش سرراست استفاده کند. می‌توان محاسبه اطلاعات متقابل میانگین باقیمانده بعد از ادغام را در زمان ثابت و مستقل از V انجام داد.

فرض کنید که V-k ادغام انجام داده‌ایم که به کلاس‌های $C_k(1), C_k(2),\ldots,C_k(k)$ منتج شده است. و اکنون می‌خواهیم ادغام $C_k(i)$ با $C_k(j)$ با برای $1\leq i\leq j\leq k$ بررسی کنیم. فرض کنید:
$P_k(l,m) = Pr(C_k(l), C_k(m))$

یعنی احتمال اینکه یک کلمه در کلاس $C_k(m)$ بعد از یک کلمه در کلاس $C_k(l)$ بیاید. فرض کنید:

$$pl_k(l) = \sum_m p_k(l,m),$$

$$pr_k(m) = \sum_l p_k(l,m),$$

$$q_k(l,m) = p_k(l,m)\log\frac{p_k(l,m)}{pl_k(l)pr_k(m)}.$$

اطلاعات متقابل میانگیت بعد از V-k ادغام برابر است با:

$$I_k = \sum_{l,m} q_k(l,m).$$

از i+j برای نشان دادن خوشه‌ای استفاده می‌کنیم که از ادغام $C_k(i)$ و $C_k(j)$ بدست آمده است. لذا برای مثال: $p_k(i+j, m) = p_k(i,m) + p_k(j,m)$ و

$$q_k(i+j,m) = p_k(i+j,m)\log\frac{p_k(i+j,m)}{pl_k(i+j)pr_k(m)}.$$

اطلاعات متقابل میانگین بعد از اینکه $C_k(i)$ و $C_k(j)$ را ادغام کنیم برابر است با:

$$\begin{aligned} I_k(i,j) &= I_k - s_k(i) - s_k(j) + q_k(i,j) + q_k(j,i) + q_k(i+j,i+j) \\ &\quad + \sum_{l\neq i,j} q_k(l,i+j) + \sum_{m\neq i,j} q_k(i+j,m), \end{aligned}$$

که

$$s_k(i) = \sum_l q_k(l,i) + \sum_m q_k(i,m) - q_k(i,i).$$

اگر I_k, $s_k(i)$, $s_k(j)$ را بدانیم آنگاه اکثر زمانی که برای محاسبه $I_k(i,j)$ لازم است صرف محاسبه مجموع‌های خط دوم معادلات بالا می‌شود. هر کدام از این مجموع‌ها تقریباً V-k عبارت دارند و لذا مسأله محاسبه $I_k(i,j)$ را از مرتبه V_2 به مرتبه V کاهش دادیم. این امر را می‌توان باز هم بهبود داد با نگه داشتن رد آن زوجهای m , l ای که برای آنها $p_k(l,m)$ مخالف صفر است.

با سنجش همه جفت‌ها، می توانیم جفتی را بیابیم که هدر رفتن اطلاعات متقابل میانگین برای آن کمترین است: $L_k(i,j) \equiv I_k - I_k(i,j)$. این مرحله را با ادغام $C_k(i)$ و $C_k(j)$ و تشکیل خوشه جدید $C_{k-1}(i)$ تکمیل می‌کنیم. اگر $j \neq k$، $C_k(k)$ را به $C_{k-1}(j)$ تغییرنام می‌دهیم و برای $l \neq i,j$ ، $C_{k-1}(l)$ را به $C_k(l)$ منتسب می‌کنیم. به وضوح $I_{k-1} = I_k(i,j)$. مقادیر $p_{k-1}, pl_{k-1}, pr_{k-1}, q_{k-1}$ را می‌توان به راحتی از p_k, pl_k, pr_k, q_k بدست آورد. اگر l و m هر دو نشان‌دهنده اندیس‌هایی باشند که هیچ کدام از آنها مساوی i یاز نباشند، آنگاه به سادگی می‌توان گفت:

$$
\begin{aligned}
s_{k-1}(l) &= s_k(l) - q_k(l,i) - q_k(i,l) - q_k(l,j) - q_k(j,l) + q_{k-1}(l,i) + q_{k-1}(i,l) \\
s_{k-1}(j) &= s_k(k) - q_k(k,i) - q_k(i,k) - q_k(k,j) - q_k(j,k) + q_{k-1}(j,i) + q_{k-1}(i,j) \\
L_{k-1}(l,m) &= L_k(l,m) - q_k(l+m,i) - q_k(i,l+m) - q_k(l+m,j) - q_k(j,l+m) \\
&\quad + q_{k-1}(l+m,i) + q_{k-1}(i,l+m) \\
L_{k-1}(l,j) &= L_k(l,k) - q_k(l+k,i) - q_k(i,l+k) - q_k(l+k,j) - q_k(j,l+k) \\
&\quad + q_{k-1}(l+j,i) + q_{k-1}(i,l+j) \\
L_{k-1}(j,l) &= L_{k-1}(l,j)
\end{aligned}
$$

در نهایت باید $s_{k-1}(i)$ و $L_{k-1}(l,i)$ را محاسبه کنیم. لذا کل فرایند بروزرسانی نیاز به محاسباتی از مرتبه V^2 دارد که در طی آن زوج‌های دیگری که باید ادغام شوند را تعیین می‌کنیم. این الگوریتم بنابراین از مرتبه V^3 است.

از Sticky Pairs و Semantic Classes هم به عنوان روشهای آماری می‌توان استفاده نمود.

4. سیستم‌های بازیابی مستندات گفتاری (spoken documents retrieval) دسته‌ای از سیستم‌های بازیابی اطلاعات هستند که کار بازیابی را بر روی فایل‌ها و مستندات حاوی گفتار انجام می‌دهند. نحوه کار این سیستم‌ها را شرح دهید و توضیح دهید که برای طراحی یک سیستم بازیابی مستندات گفتاری چه اجزایی لازم است. نحوه کار هر یک از این اجزا را شرح دهید.

هدف از سیستم‌های بازیابی مستندات گفتاری فراهم آوردن بازیابی محتوی-محور عبارات و بیانات از آرشیوهای ضبط شده گفتاری است. بنابراین SDR معادل بازیابی محتویات گفتاری در فایل‌های صوتی دیجیتال است که بر اساس یک پرس و جو[7]ی متنی مرتب شده اند. معمولاً این پرس و جو شامل دنباله‌ای نوع‌دار از کلمات، یک سوال، یا گزاره‌ای از اطلاعات مورد نیاز است. مستندات گفتاری، ضبطهای گفتاری صحبت انسان است که قبلاً ایندکس شده‌اند و توسط یک سیستم بازشناسی گفتار، به طور خودکار آوانویسی شده است. لذا هدف SDR نوعی از بازیابی اطلاعات است؛ یعنی یافتن خودکار آن مستندات کامل یا گزیده‌ای است که یا شامل کلمات و عبارات موجود در آن پرس و جو هستند یا از لحاظ معنایی با اطلاعات خواسته شده مرتبط می‌باشند. بنابراین آوانویسی گفتار باید خودکار باشد و بازیابی باید براساس پرس و جوی کاربر باشد. بازیابی موسیقی، ترانه یا صداهای غیرزبانی در محدوده بازیابی مستندات گفتاری نمی‌گنجد. سوال و جواب مختصر که در آن کاربر منتظر این است تا جواب کوتاهی در مقابل سوالش بشنود (به جای اینکه بخواهد یک مستند گفتاری کامل داشته باشد) نیز در تعریف SDR نمی‌گنجد.

هدف از توسعه مکانیزمی که دسترسی به اطلاعات گفتاری را فراهم می‌آورد نسبتاً واضح است. دردسترس‌بودن کامپیوترهای ارزان، دستگاه‌های ذخیره‌سازی و ظرفیت‌های با پهنای باند بالا برای انتقال اطلاعات منجر به مجموعه‌های چندرسانه‌ای بزرگ شده است. مردم عادت کرده‌اند که بطور مجازی به همه اطلاعات متنی درون اینترنت دسترسی داشته باشند. بدون SDR، دسترسی به آرشیوهای صوتی یا حداقل مجموعه‌های صوتی گفتاری محدود به آن دسته از مستنداتی می‌شود که به صورت دستی آوانویسی شده‌اند یا با کلمات کلیدی ایندکس گذاری شده‌اند. علیرغم اینکه درصد قابل توجهی از برنامه‌های رادیو تلویزیونی امروزه دارای متونی هستند که بطور دستی تهیه شده‌اند یا حداقل طرح کلی تقریبی متن را دارند، درصد بسیار بیشتری از ضبطهای گفتاری بدون متن هستند چرا که هزینه متن‌نویسی توسط انسان بسیار بالا است یا مربوط به برنامه قدیمی‌تر رادیو تلویزیون می‌شوند که متون آنها یا گم شده یا اصلاً تهیه نشده بوده است.

SDR یک قابلیت جستجو و بازیابی کامل به صورت آنچه که امروزه برای محتویات متنی موجود است، فراهم می‌آورد.این قابلیت در کاربردهای زیر مفید است:

- جستجو درون متون کنفرانس‌های ویدیویی
- دسترسی به قسمت‌هایی از دروس آموزشی ضبط شده
- یافتن محتویات خاص در صوت یا تصویر آموزشی
- سازماندهی ایمیل صوتی آرشیوشده با محتویات گفتاری
- دسترسی به خبرهای مورد نظر از تلویزیون یا رادیو

[7] Query

- ذخیره‌سازی جلسات به صورت مستندات
- بازیابی متون صوتی از برنامه‌های ورزشی و تفریحی، شامل جُنگ‌ها، فیلم‌ها، برنامه‌های پرسش و آزمون و غیره.

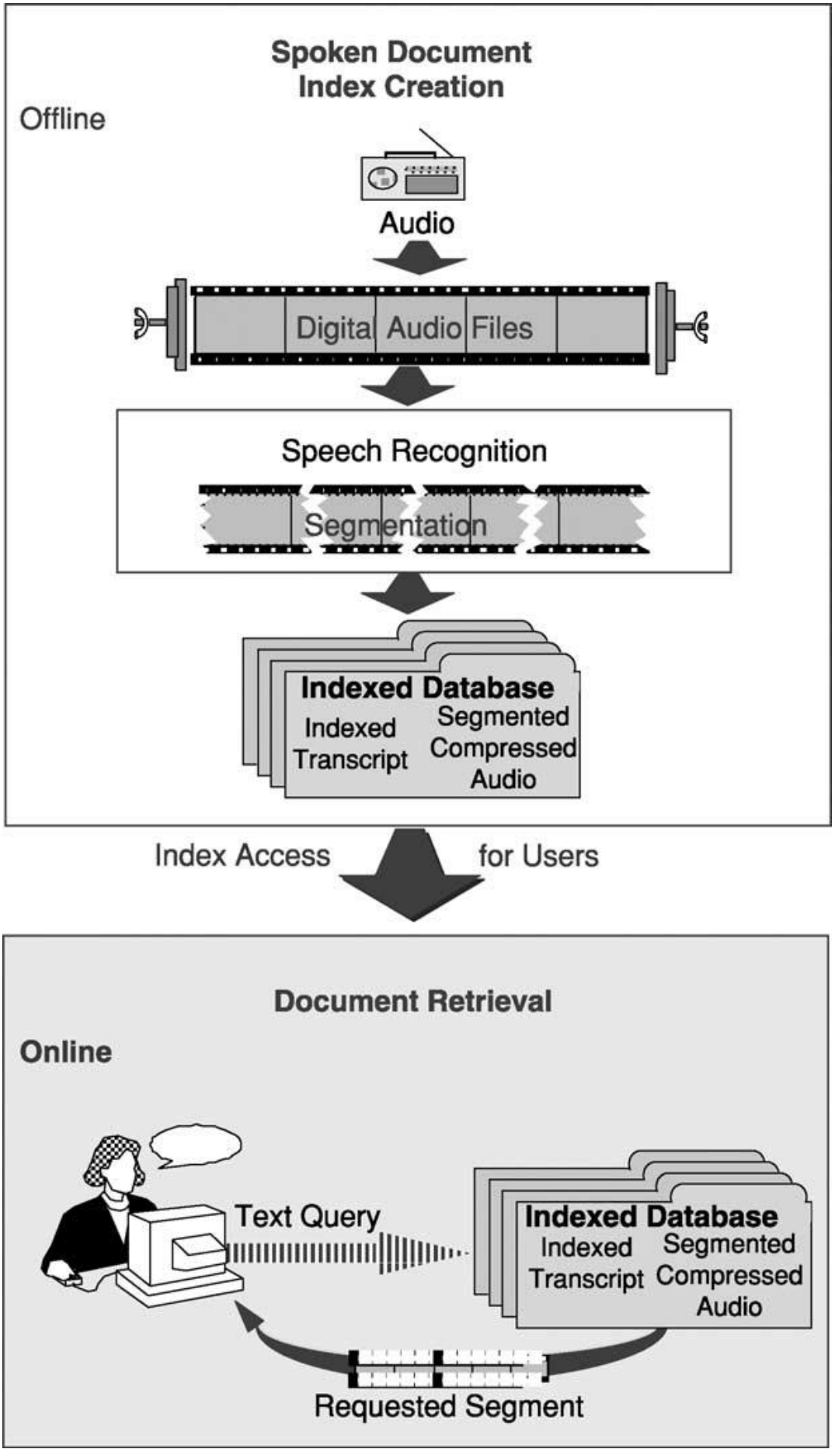
در مرحله اول، سیستم SDR یک متن آوانویسی‌شده از صوت‌های ضبط شده تولید می‌کند تا بازیابی مبتنی بر متن را روی مستندات صوتی امکان‌پذیر سازد. ایده اصلی این است که ابتدائاً بازشناسی گفتار خودکار را روی مستندات گفتاری اعمال سازد تا یک متن آوانویسی شده بدست آید. واحدهای آوانویسی معمولا کلمه هستند؛ هرچند هجاها و واج‌ها نیز بکار رفته‌اند. هر کلمه در متن با زمان وقوع آن نشانه‌گذاری می‌شود. معمولاً SDR طوری طراحی می‌شود که آرشیوهای وسیعی از مستندات گفتاری را که قبلاً ایندکس گذاری شده بودند، مورد جستجو قرار دهد. در مورد مجموعه‌های پایدار صوتی، بازشناسی گفتار یکبار قبل از اینکه آرشیو ایندکس‌گذاری شود انجام می‌شود. در آرشیوهایی که به صورت فعال مستندات گفتاری را جمع‌آوری می‌کنند، بازشناسی گفتار به صورت پیوسته و همزمان با ذخیره صوت انجام می‌شود.

مرحله بعدی در فرایند SDR ساختن ایندکس بازیابی اطلاعات است. برای آسانی تطبیق کلمات ریشه با تصریف‌های مختلف، یک فرایند ریشه‌یابی همه کلمات تشخیص داده شده را به پایه‌ها و ریشه‌های آنها تبدیل می‌کند. بعد از حذف کلماتی که بسیار زیاد رخ می‌دهند، یا stop word ها، باقی ریشه‌های کلمات به صورت یک ایندکس معکوس در می‌آیند که دسترسی سریع بر اساس کلمات پرس و جو را ممکن می‌سازد. هر ریشه کلمه شامل یک ورودی در ایندکس است، که همه مستنداتی که در آن رخ داده است، تعداد دفعاتی که در مستندات رخ داده است و زمان هر رخداد واحد را لیست می‌کند. این آمار بر اساس توزیع کلمات درون مستندات به منظور رتبه‌بندی تشابه بازیابی، توزیع می‌شود.

وقتی آرشیو ایندکس گذاری شده مورد دسترسی واقع می‌شود، یک پرس و جو که نمایانگر اطلاعات مورد نیاز کاربر است به صورت دنباله‌ای از کلمات یا جملات کامل بیان می‌شود. علیرغم اینکه پرس و جوها معمولاً تایپ می‌شود، در برخی موارد، پرس و جو ممکن است به صورت صوتی گفته شود. در مرحله پردازش پرس و جو، املای پرس و جو با همه نقطه‌گذاری‌ها نرمال‌سازی می‌شود. لذا برای مثال بعد از نرمال‌سازی، ریشه‌یابی و حذف stop word ها، پرس و جوی "C.I.A's use of spy planes in 1985?" بصورت "CIA SPY PLANE NINETEEN EIGHTY FIVE" درمی‌آید. این پرس و جو سپس به صورت دنباله‌ای(برداری) از واحدهایی با نوعی یکسان با واحدهای بازشناسی ایندکس شده (یعنی کلمات، هجاها یا واج‌ها) تبدیل می‌شود.

برای بازیابی مستندات گفتاری مرتبط با پرس و جوی کاربر، هر مستند در آرشیو به طور خودکار با بردار پرس و جو مقایسه می‌شود تا مشخص شود چقدر این مستند با پرس و جو مطابقت دارد. تابع مطابقت[8] R(Q, D) ارتباط (R) یک پرس و جو (Q) را با یک مستند (D) محاسبه می‌کند. لیستی از مستندات گفتاری مطابق با پرس و جو به کاربر برگردانده می‌شود که در آن مکان کلمات مورد مطابقت مشخص شده است. لیست مستندات گفتاری بر اساس تشابه کاهشی بین پرس و جو و محتوای مستند مرتب سازی می‌شود. یک فرایند نوعی SDR در شکل زیر نشان داده شده است:

[8] Matching function


Spoken Document
Index Creation
Offline
Audio
Digital Audio Files
Speech Recognition
Segmentation
Indexed Database
Indexed Transcript
Segmented Compressed Audio
Index Access
for Users
Document Retrieval
Online
Text Query
Indexed Database
Indexed Transcript
Segmented Compressed Audio
Requested Segment

5. برای ارزیابی سیستم‌های ترجمه ماشینی روش‌های مختلفی پیشنهاد شده است. این روش‌ها را با جزئیات شرح دهید.

روش‌های ارزیابی ترجمه ماشینی را می‌توان از یک دیدگاه کلی به دو دسته ارزیابی انسانی و ارزیابی خودکار تقسیم کرد. ابتدا روشهای انسانی را بررسی می‌کنیم.

یکی از مولفه‌های اصلی **روش ALPAC** مطالعه سطوح مختلف ترجمه انسانی با خروجی ترجمه ماشینی و استفاده از سوژه‌های انسانی برای قضاوت بود. دو متغیر در این روش درنظر گرفته می‌شود:
وفاداری (یا دقت): جمله ترجمه شده در مقایسه با جمله اصلی شامل چه میزان اطلاعاتی بود (طبق مقیاس 0-9)
قابلیت فهم: ترجمه خودکار چقدر قابل فهم و درک است (طبق مقیاس 1-9).

در **روش ARPA**، معیارهای ارزیابی عبارتند از:

- ارزیابی فهم[9]: این روش که اطلاع‌دهندگی[10] نیز نامیده می‌شود بدین منظور است که مستقیماً سیستم‌ها را براساس نتایج آزمون‌های فهم چندگزینه‌ای مقایسه کند. بنابراین این روش، یک معیار ارزیابی برونگرا[11] است. کیفیت ترجمه ماشینی به صورت غیرمستقیم با استفاده از سوژه‌های انسانی بررسی می‌شود بطوری که متن‌های ترجمه شده خودکار را می‌خوانند و به چندین سوال در رابطه با آن پاسخ می‌دهند.
- ارزیابی پانل کیفیت[12]: در این روش، ترجمه‌های به یک پانل از متکلمین بومی خبره داده می‌شود که مترجمان حرفه‌ای باشند.
- کفایت و روانی[13]: گروهی از سوژه‌های انسانی برای قضاوت روی مجموعه‌ای از ترجمه‌های یک یا چند متن مورد نیاز است. به داوران، قسمتی از یک ترجمه نشان داده می‌شود و از آنها خواسته می‌شود به دو معیار کفایت و روانی آن ترجمه رأی دهند.

در **روش نگهداری معنا**[14] معنی ترجمه با معنی مبدأ مقایسه می‌شود.
روش زمان خواندن[15]: زمان خواندن یعنی زمانی که یک کاربر نیاز دارد تا متنی را بخواند و احساس کند به اندازه کافی آنرا فهمیده است.
روش نیاز به پس-ویرایش[16]: حداقل تعداد فشردن کلید (موقع تایپ) که لازم است یک ترجمه به ترجمه معتبری تبدیل شود.
روش زمان پس-ویرایش[17]: زمان لازم برای تبدیل ترجمه خودکار به یک ترجمه معتبر.
روش آزمون Cloze: یک آزمون خوانایی که در آن توانایی کاربر برای پر کردن جاهای خالی که عامدانه از متن ترجمه شده حذف شده‌اند مورد بررسی قرار می‌گیرد.

---

[9] Comprehension
[10] Informativeness
[11] Extrinsic
[12] Quality Panel Evaluation
[13] Adequacy And Fleuncy
[14] Meaning Maintanace
[15] Read Time
[16] Required Post-Editing
[17] Post-Edit Time

روش وضوح[18]: از افراد خواسته می‌شود که وضوح هر جمله را بر اساس مقیاس 1 تا 3 درجه بندی کنند.

حال به مرور روشهای خودکار می‌پردازیم:
روش‌های مبتنی بر فاصله ویرایش[19]:

- **WER**[20]: نرخ خطای کلمه. این معیار براساس فاصله لونشتاین کار می‌کند یعنی حداقل تعداد تعویض، حذف یا درج که باید انجام شود تا ترجمه خودکار را به یک ترجمه معتبر تبدیل نمود.
- **PER**[21]: نرخ خطای کلمه مستقل از موقعیت. یک نقطه ضعف روش WER این است که بازمرتب‌سازی کلمات را امکان‌پذیر نمی‌سازد. برای حل این مشکل، روش PER کلمات موجود در دو جمله را بدون توجه به ترتیب آنها مقایسه می‌کند.
- **TER**[22]: نرخ ویرایش ترجمه. TER میزان پس-ویرایش‌هایی که لازم است یک فرد انجام دهد تا خروجی سیستم را مطابق با ترجمه مرجع کند، اندازه می‌گیرد.

روش‌های مبتنی بر دقت[23]: این روشها دقت واژگانی را محاسبه می‌کنند، یعنی نسبت واحدهای واژگانی (نوعاً n-gram هایی با اندازه متغیر) در ترجمه خودکار که توسط ترجمه مرجع پوشش داده می‌شوند:

- **BLEU**[24]: این روش ngram های دقت واژگانی را تا ngram های به طول 4 محاسبه می‌کند.
- **NIST**[25]: نسخه بهبودیافته‌ای از BLEU. مهمترین تفاوت آن با BLEU در این است که چگونه امتیازهای ngram را میانگین‌گیری کنند. BLEU روی میانگین هندسی متمرکز است ولی NIST یک میانگین حسابی را در نظر می‌گیرد. علاوه براین NIST ngram های تا طول 5 را در نظر می‌گیرد.
- **WNM**: نسخه‌ای از BLEU که ngramها را براساس برجستگی آماری که از یک پیکره تک‌زبانی عظیم تخمین زده شده است، محاسبه می‌کند.

روش های مبتنی بر یادآوری[26]: این روش‌ها یادآوری واژگانی را محاسبه می‌کنند یعنی نسب واحدهای واژگانی در ترجمه مرجع به ترجمه خودکار.

- **ROUGE**[27]: یادآوری واژگانی را بین ngram های تا طول 4 محاسبه می‌کند. همچنین در این روش می‌توان از ریشه‌یابی و تطابق غیرپیوسته (skip bigrams) استفاده نمود.
- **CDER**[28]: نرخ خطای پوشش/گسسته. این روش بازمرتب سازی بلوک‌ها را مدل می‌سازد. و مبتنی بر فاصله CDCD است که توسط لپرستی و تامکینز معرفی شده است.

---

[18] Clarity
[19] Edit Distance
[20] Word Error Rate
[21] Position-Independent Word Error Rate
[22] Translation Edit Rate
[23] Precision Oriented
[24] Bilingual Evaluation Understudy
[25] National Institute Of Standards And Technology
[26] Recall Oriented
[27] Recall Oriented Understudy For Gisting Evaluation
[28] Cover/Disjoint Error Rate

روش‌های ترکیب کننده دقت و یادآوری:

- **GTM**: یک F-measure است.
- **METEOR**: یک F-measure که بر اساس برهم‌نهی unigramها کار می‌کند.
- **BLANC**: خانواده‌ای از ngram های پویای قابل یادگیری.
- **SIA**[29]: برهم‌نهی تکراری تصادفی. روشی مبتنی بر برهم‌نهی سست دنباله‌ها و تطابق تصادفی کلمات و برهم‌نهی تکراری.

دو مسأله مهم در معیارهای ارزیابی خودکار ترجمه ماشینی عبارتند از: کمبود ویژگی‌های به اندازه کافی قوی برای دستیابی به ساختار سطح جمله و معنی مبدأ؛ و مشکل بودن طراحی توابع مناسبی که بتوانند این ویژگی‌ها را به یک الگوریتم ارزیابی کیفیت ترجمه تبدیل کنند. با اینکه روش‌های ارزیابی ترجمه ماشینی، همبستگی‌های خوبی با قضاوت‌های انسانی از نقطه نظر کفایت و روانی (سلیس بودن) نشان می‌دهند، هنوز جا برای بهبود وجود دارد. مجموعه داده‌هایی که برای ارزیابی خودکار ترجمه ماشینی استفاده می‌شود هنوز کوچک هستند و بازه زبان‌هایی که این قضاوت‌ها برای آنها موجود است هنوز محدود می‌باشد.
باید بین دو مولفه ارزیابی ترجمه ماشینی تمایز قائل شد: آمار محاسبه شده روی ترجمه‌های کاندید و مرجع و تابع استفاده شده در تعریف معیارهای ارزیابی و تولید امتیازات ترجمه. بیشترین آمار استفاده شده شامل موارد زیر است:

- Bag-of-word overlap
- Edit distance
- Longest common subsequence
- Ngram overlap
- Skip-bigram overlap

بیشترین توابع استفاده شده شامل ترکیبات مختلفی از دقت و یادآوری[30] می‌باشد که شامل weighted precision و F-measure است.
کار اولیه روی ارزیابی کیفیت ترجمه روی معیارهای مبتنی بر فاصله[31] متمرکز بود. در حوزه ترجمه ماشینی، فاصله ویرایش[32] (Levenshtein) نشاندهنده تعداد درج، حذف و تعویض‌های لازم است که یک ترجمه کاندید را به یک ترجمه مرجع تبدیل کند. یک معیار دیگر ارزیابی مبتنی بر فاصله ویرایش، نرخ خطای کلمات[33] می‌باشد که فاصله ویرایش نرمال‌شده را محاسبه می‌کند.
Bleu یک معیار ارزیابی دقت وزندار است که توسط IBM معرفی شد. به خاطر تأثیری که این روش روی حوزه ترجمه ماشینی گذاشت، Bleu و گسترش‌های آن بعنوان یکی از کاربردی‌ترین معیارهای ارزیابی ترجمه ماشینی به حساب آورده می‌شوند. ارزیابی‌های اخیر ترجمه ماشینی یک نسخه از Bleu که توسط NIST توسعه داده شده است را نیز بکار گرفته‌اند. هر دو روش مبتنی بر ویژگی‌های موضعی[34](ngrams) هستند و صریحاً از ویژگی‌های سطح جمله استفاده نمی‌کنند. General Text Matcher(GTM) روشی دیگر برای ارزیابی ترجمه ماشینی می‌باشد که به ngram های بزرگتر بها می‌دهد؛ به جای اینکه مانند روش محاسبه Bleu وزنهای برابر به آنها بدهد.

[29] Stochastic Iterative Alignment
[30] Precision And Recall
[31] Distance-Base
[32] Edit Distance
[33] Word Error Rate
[34] Local

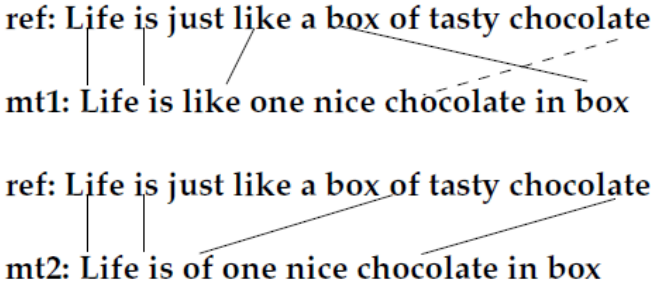
Rouge-L یک معیار ارزیابی است که مبتنی بر بزرگترین زیردنباله مشترک (LCS) می‌باشد. ایده اصلی استفاده از LCS برای ارزیابی خودکار ترجمه ماشینی این است که زیردنباله‌های مشترک طولانی، همپوشانی بیشتری بین یک ترجمه کاندید و یک ترجمه مرجع نشان می‌دهند. با استفاده از LCS، Rouge-L برای ترجمه کاندید و ترجمه مرجع، دقت و بازیابی تعریف می‌کند و F-measure متناظر را بدین گونه حساب می‌کند:

$$P_{lcs} = \frac{LCS(cand, ref)}{|cand|} \quad (1)$$

$$R_{lcs} = \frac{LCS(cand, ref)}{|ref|} \quad (2)$$

$$F_{lcs} = \frac{(1+\beta^2) P_{lcs} R_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (3)$$

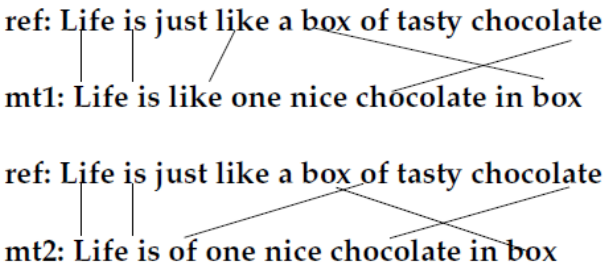
به همین ترتیب روش Rouge-W از طولانی‌ترین زیردنباله مشترک استفاده می‌کند ولی وزنهای بالاتری به دنباله‌هایی که شکاف‌های کمتری دارند می‌دهد. این امر به Rouge-W این امکان را می‌دهد که تأکید و تمرکز بیشتری روی به هم پیوستگی[35] داشته باشد و معنی موضعی را همزمان با ساختار سطح جمله به نحو بهتری بدست آورد. با این حال این معیار هنوز تمایز قائل نمی‌شود بین ترجمه‌هایی که LCS وزندار مشابه دارند ولی تعداد زیردنباله‌های کوچکتر متفاوتی دارند که نشاندهنده همپوشانی اضافی است که توسط Rouge-W بدست نیامده است. Rouge-S با ترکیب دقت/یادآوری skip-bigramهای ترجمه کاندید وترجمه مرجع، سعی بر تصحیح این مشکل دارد. هرچند با استفاده از skip-bigramهای با وزن یکسان، Rouge-S معنی موضعی ناهمسان‌تری[36] و ساختار سطح جمله بیشتری بدست می‌آورد. علاوه بر این، با استفاده از skip-bigramها ممکن است نتوانیم اطلاعات موردنیاز درباره ساختار سطح بالاتر موجود را بدست آوریم. مثالی از تطبیق[37] برای Rouge-W در شکل زیر آمده است:



روش همه زیردنباله‌های مشترک[38] یا ACS [که معادل با یافتن همه skip-ngram های مشترک است که یک skip-ngram بصورت یک زیردنباله از کلمات تعریف می‌شود] سعی بر این دارد که نشانه‌های[39] جمله موضعی و سراسری را با محاسبه همپوشانی بین ترجمه‌های کاندید و مرجع با استفاده از skip-ngram های وزندار بدست آورد.

[35] Coherence
[36] Less Coherent
[37] Alignment
[38] All Common Subsequences
[39] Cues

روش Meteor یک معیار مبتنی بر unigram است که تطبیق یکنواخت[40] کلمات بین خروجی ترجمه ماشینی و مرجع‌ها را به تطبیق کلمات متقاطع جریمه‌ای[41] ترجیح می‌دهد. دو مسأله در رابطه با Meteor وجود دارد. اول اینکه شکاف‌های موجود در کلمات تطبیق‌شده را درنظر نمی‌گیرد که خود ویژگی مهمی برای ارزیابی سلیس بودن جمله است؛ و دوم اینکه نمی‌تواند از چندین مرجع بصورت همزمان استفاده کند. Rouge و Meteor هر دو از Wordnet و ریشه‌یاب Porter برای افزایش احتمال تطابق خروجی ترجمه ماشینی با کلمات مرجع استفاده می‌کنند. این پردازش‌های صرفی و استخراج مترادفها برای زبان انگلیسی موجود هستند ولی لزوماً برای بقیه زبان‌ها اینگونه نیست. مثالی از تطابق در Meteor در شکل زیر آمده است:



[40] Monotonic
[41] Penalizing Crossing Word Alignments